



Escuela Técnica Superior
de Ingeniería Industrial

Programación de CompactRIO para la medida y adquisición de datos en cuadros eléctricos de BT

Titulación: Ingeniero Industrial
Intensificación: Sistemas Eléctricos
Alumno/a: Antonio Enrique Moreno Carraco
Director/a/s: Ángel Molina García

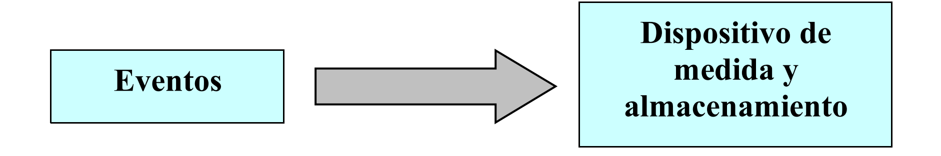
Cartagena, 22 de Diciembre de 2009

Índice de Contenidos

Índice de Figuras

Objetivos

Necesitamos tener un sistema flexible de adquisición de datos en el que una de las aplicaciones sea la captura de huecos de tensión. Este dispositivo debe funcionar de forma autónoma y disponer de memoria suficiente como para almacenar todos los datos a guardar generados un periodo de tiempo amplio. En el mercado podemos encontrar dispositivos capaces de satisfacer estos objetivos el gran problema en unos casos es su elevado precio y en otros con un precio razonable es su falta de flexibilidad.



El dispositivo que mejor se adapta a estos objetivos con un precio asequible y una calidad contrastada es el CompactRIO de National Instruments, que al ser totalmente configurable tiene una gran flexibilidad, aunque precisamente esta cualidad conlleva un mayor trabajo de programación. Además tiene un precio razonable para lo que cuesta este tipo de dispositivos e incorpora una amplia memoria para almacenar datos (2 GB).

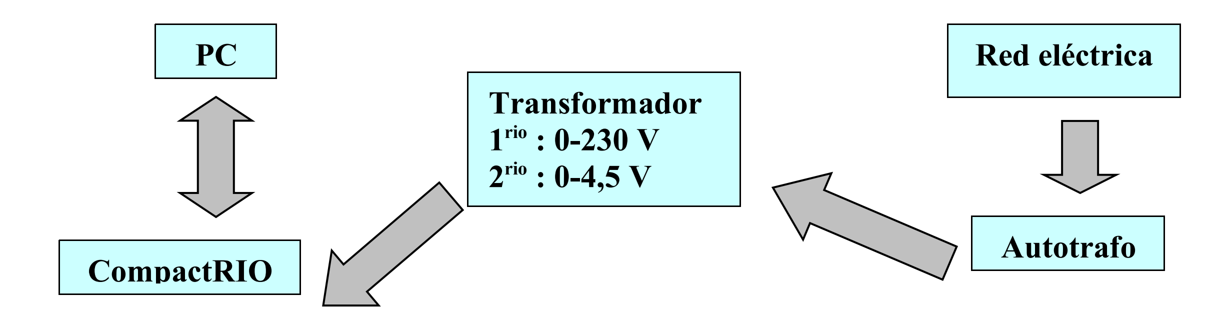


Figura 1: CompactRIO.

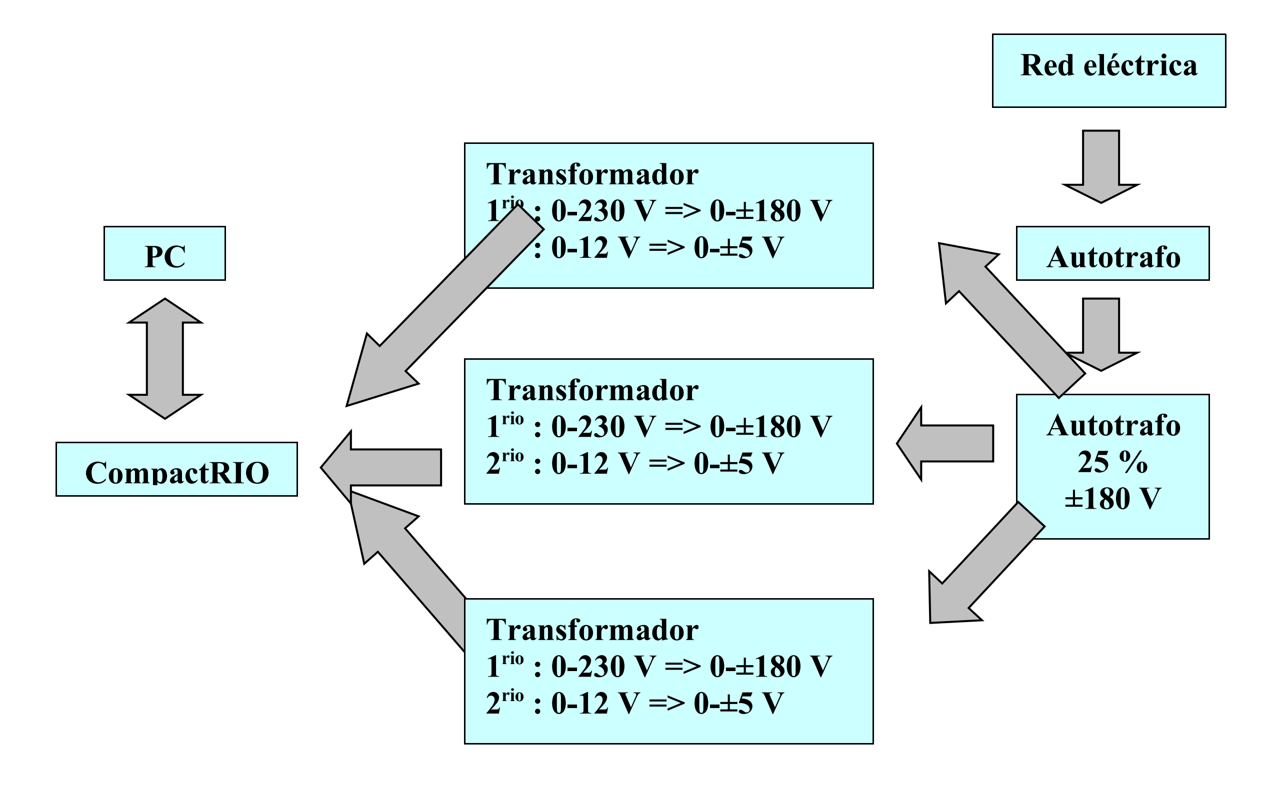
Equipamiento de laboratorio

a) Un CompactRIO modelo cRIO-9014 de ocho ranuras.
b) Dos tarjetas de adquisición de datos NI 9215 con "Screw Terminal", que quiere decir que posee cuatro canales de dos pines.
c) Un PC.
d) Dos autotrafo trifásico.
e) Un transformador de 0-125-220 / 0-4,5-6-7,5-9 (V).
f) Tres transformadores 0-125-220 / 12-0-3-12 (V).
g) Dos polímetros.
h) Un generador de funciones.
i) Un osciloscopio.

Esquema del montaje para los ensayos monofásicos:



Esquema del montaje para los ensayos trifásicos:



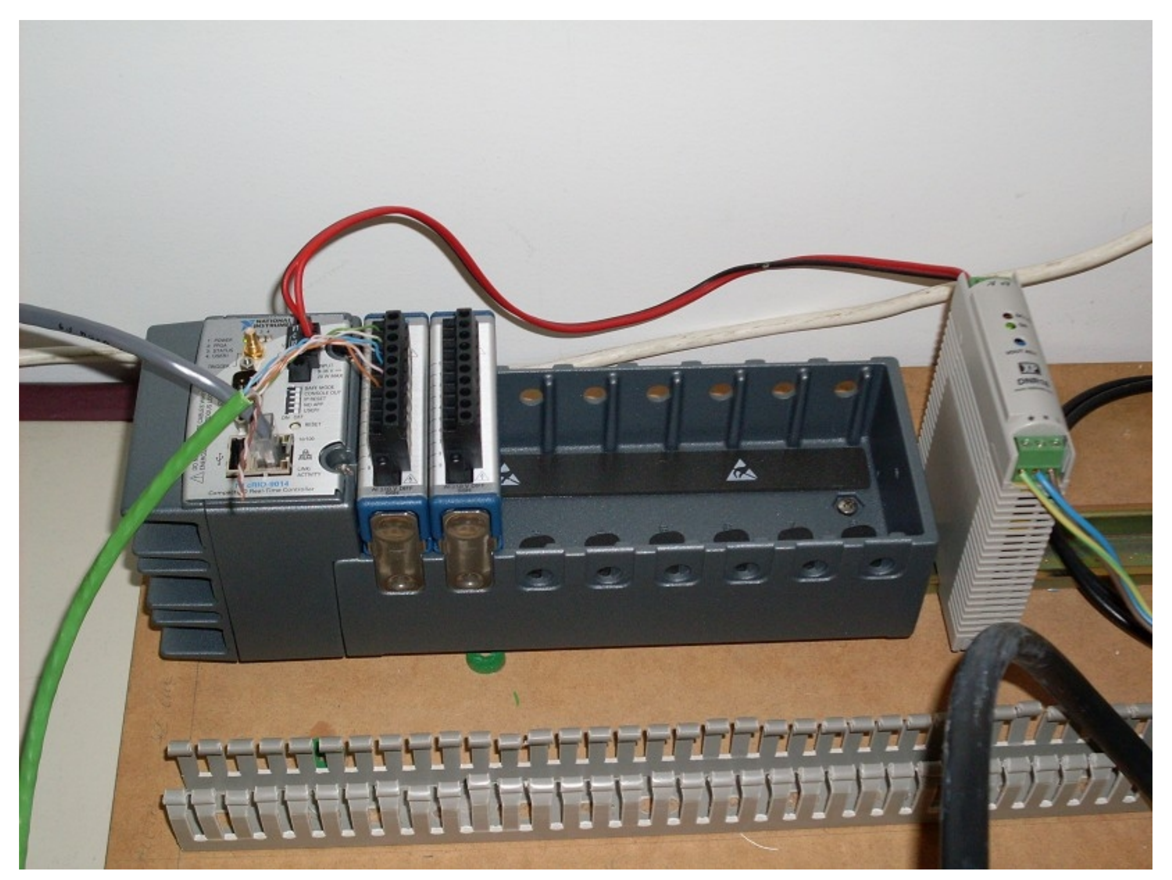


Figura 2: CompactRIO y fuente de alimentación de nuestro montaje.

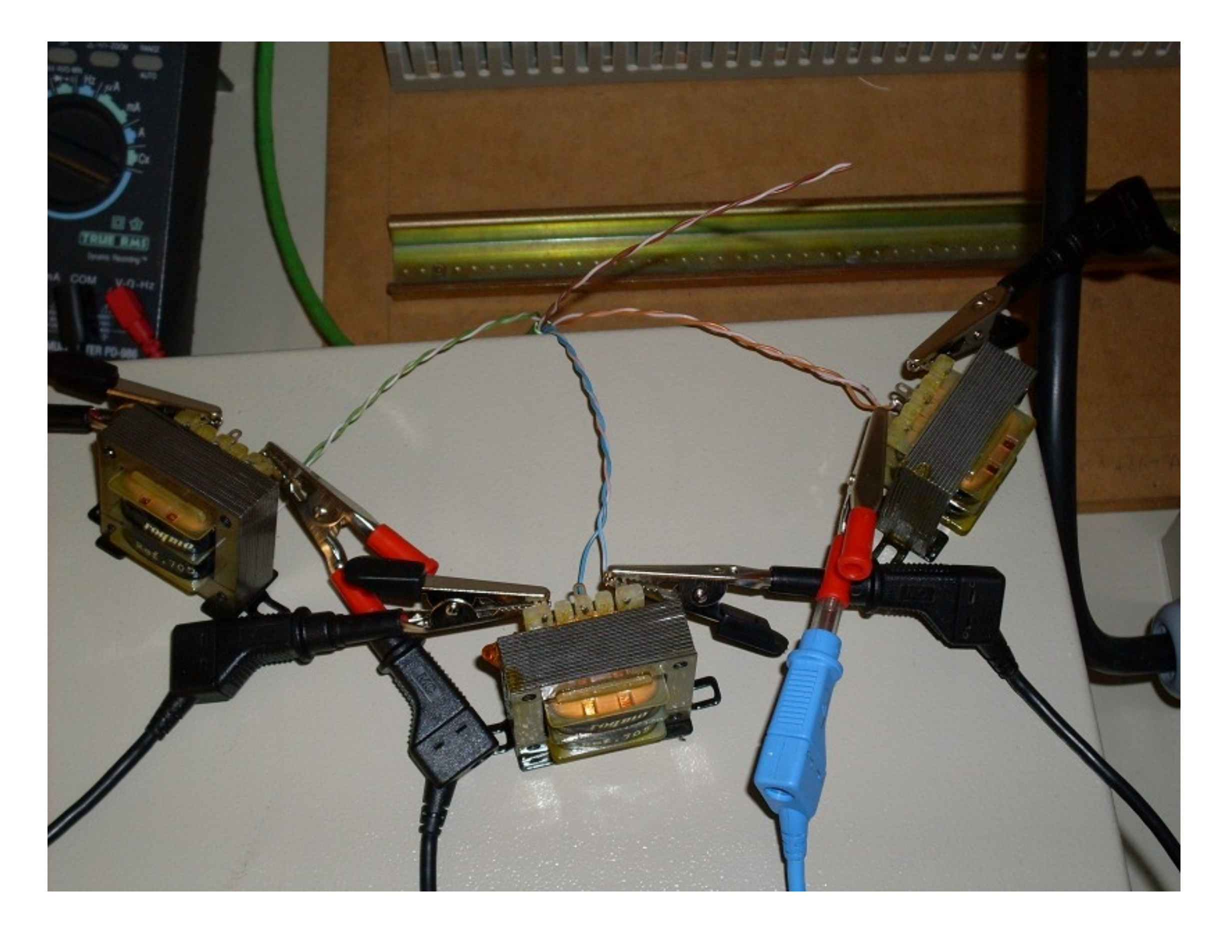


Figura 3: Tres transformadores para el montaje trifásico.

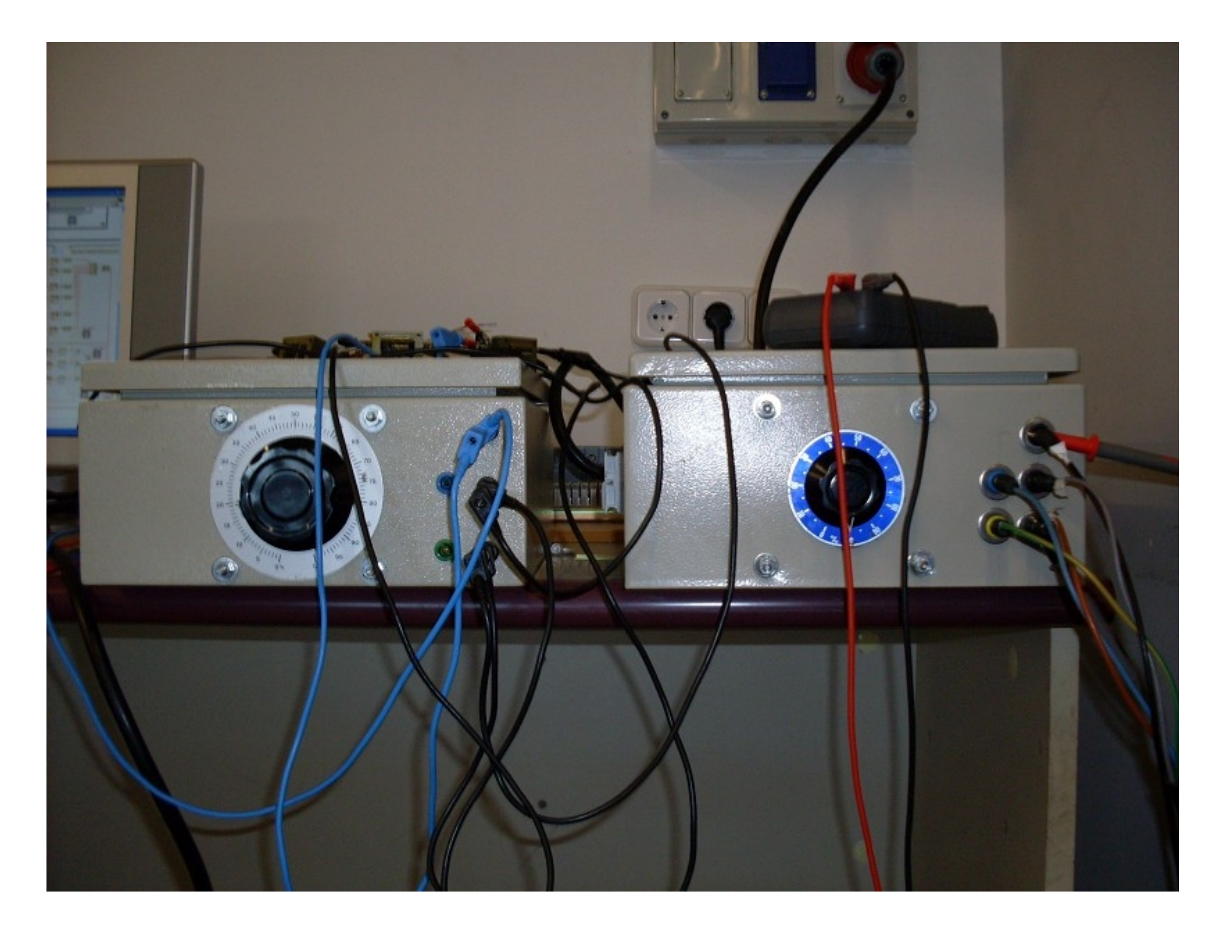


Figura 4: Dos autotrafos colocados en serie para el montaje trifásico.

Configuración del Compact RIO

El Compact RIO consta de un chasis con una pequeña CPU (Figura 7) y de una serie de tarjetas de adquisición de datos (Figura 5), en nuestro caso de dos tarjetas (Figura 6). El chasis que vamos a utilizar admite como máximo la conexión de ocho tarjetas. Una vez unidos el chasis y los dos módulos hay que alimentados mediante una fuente de continua de 18 W y 24 VDC. Ahora debemos configurar el Compact RIO para poder comunicarnos con él. La forma de hacerlo es a través de su puerto LAN (de red), bien directamente por red local, o bien por Internet. Hemos decidido que lo vamos a hacer conectando una PC directamente al Compact RIO (con un cable RJ-45).



Figura 5: Tarjeta de adquisición de datos.

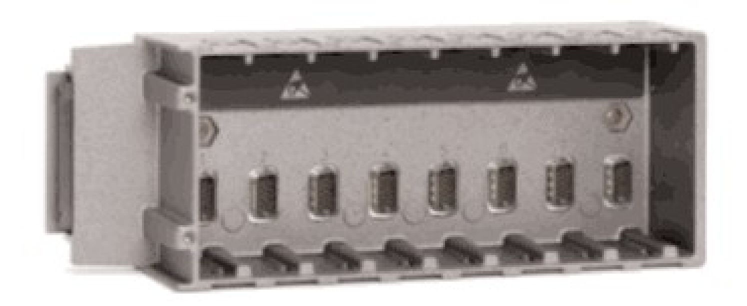


Figura 6: Chasis del CompactRIO.

Para poder programar el Compact RIO, antes este sistema formado por él mismo y el PC tiene que estar configurado. La herramienta que nos proporciona National Instruments para este fin es la aplicación Measurement & Automation Explorer (MAX).



Figura 7: CPU del CompactRIO.

Los pasos a seguir son los siguientes (véase la Figura 7 para entender mejor cada paso):

- Comprobar que todos los botones el Compact RIO está en la posición "OFF". Cada vez que se mueva un botón del Compact RIO hay que resetearlo, presionando el botón "RESET" que hay debajo de los botones. Si no se resetea el Compact RIO no va a ser consciente de los cambios de posición de estos botones, por lo tanto de la nueva disposición-configuración. Para cuidarnos en salud, reseteamos el Compact RIO aunque no modifiquemos ninguna posición de los botones.
- Ejecutar la herramienta MAX.
- Seleccionar nuestro Compact RIO que aparecerá debajo de la pestaña "Remote Systems". Si no aparece la causa más probable es que el cortafuegos del PC lo bloquee, por lo que habrá que desactivar esta protección de nuestro ordenador. Si estamos configurando el Compact RIO por primera vez o ha sido formateado la IP que aparecerá será la 0.0.0.0, que es la establecida por defecto. Además hasta que no se complete la configuración, es éste el nombre (0.0.0.0) con el que se le denomina a nuestro Compact RIO en el MAX.

- En nuestro caso no es necesario puesto que el compact RIO está conectado directamente al PC, pero si esta conexión fuera a través de una red local o Internet habría que comprobar que el número de serie del aparato encontrado sea el nuestro para no intentar configurar otro por error.
- Es recomendable formatear el Compact RIO en este punto, pero si no se desea hacer, saltar los pasos correspondientes al formateo. Mover el botón "SAFE MODE" a su posición "ON" y presionar el botón "RESET". Por seguridad sólo se puede formatear el Compact RIO activando el modo seguro del aparato.
- Clickear con botón derecho del ratón sobre el icono del Compact RIO del MAX, seleccionar "Format Disk" y aceptar.
- Cuando el proceso de formateo haya acabado nos saldrá una pantalla de aviso en el MAX, entonces desactivar el modo seguro del Compact RIO y resetearlo, posteriormente reiniciar el aparato.
- El manual de instrucciones del Compact RIO nos dice que la mejor comunicación con él es a través de una IP dinámica pero en la práctica no es así. Porque esta configuración da errores y después de formatear el equipo resulta imposible comunicarse con él. Cuando sucede esto sólo queda una solución, formatearlo manualmente con el botón "IP RESET" para obtener de nuevo una IP nula y comenzar de nuevo todos los pasos de la configuración.
- Seleccionar en nuestro PC la opción de "Usar la siguiente dirección IP:" dentro de la ventana "Propiedades de Protocolo Internet (TCP/IP)" que a su vez se accede a través de "Propiedades de conexión de área local" en "Conexiones de red"
  IP: 169.254.209.247
  Máscara de Subred: 255.255.0.0 (la misma que va a tener nuestro aparato)
- Obtener un a IP estática en el Compact RIO; en la ventana principal del MAX seleccionar nuestro equipo y en la ventana que se abre seleccionar "Suggest Values…". El MAX nos dará los datos correspondientes a nuestra red, es decir, nuestro ordenador. Se recomienda cambiar el último número de la IP que nos proporciona por un número alto por ejemplo 100 para evitar solapamientos debidos a la asignación dinámicas de IP´s de la red local. Para nuestro aparato tenemos una IP 192.168.1.100
- Seleccionar un nombre para el Compact RIO, en nuestro caso hemos elegido Compact RIO.
- Clickear en "Apply" y permitir que el MAX reinicie el equipo.

- Cuando se reinicie el equipo en la ventana del MAX, ya aparecerá el nombre elegido, clickear con el botón derecho sobre el icono de “Software” y seleccionar “Add/Remove Software”.
- Puesto que tenemos un equipo de gama alta con 2 GB de memoria interna, seleccionar la instalación completa para evitar posibles problemas futuros. Se instalará en el aparato el software (llamado NI-RIO) perteneciente a LabVIEW necesario para ejecutar internamente los programas VI´s y devolver los datos al PC. Previamente tanto el LabVIEW como el NI-RIO tienen que estar instalados en PC y en la versiones mínimas para que funcione nuestro Compact RIO, que son en este aparato concreto el LabVIEW 8.5.1 y el NI-RIO 2.4.1.
- Una vez finalizados estos pasos ya está el aparato configurado y podemos cerrar el MAX.

Una vez configurado la pantalla del MAX sería la siguiente:

Figura 8: Pantalla del MAX.

Programación del Software

Introducción y requerimientos

Una vez configurado el CompactRIO, con una correcta comunicación entre los hardware, es decir, entre éste y el PC. Hay que programar el CompactRIO con el software llamado LabVIEW.

El LabVIEW requerido es la versión 8.5.1. o posteriores.

Los drivers del CompactRIO y de las tarjetas de adquisición de datos pertenecen al archivo NI-RIO 2.4.1.

Las carpetas proporcionadas por la Universidad de Albacete llamadas LV RT, LVFPGA85_CD1 y por última LVFPGA85_CD2.

El programa debe satisfacer una serie de requisitos que a grandes rasgos son los siguientes:

1º) Grabar los datos medidos o permanecer a la espera según lo desee el usuario.

2º) Grabar los datos durante un tiempo y permanecer a la espera otro, ambos tiempos introducidos por el usuario.

3º) Crear un buffer de datos correspondientes a un periodo de tiempo elegido por el usuario llamado tiempo pre-evento, y evaluar el RMS del siguiente ciclo competo de onda. Ahora se puede presentar varias situaciones:

a) Que este valor de RMS sea mayor que el valor condición introducido por el usuario, por lo que se eliminará el primer ciclo del buffer de datos y se incluirá los datos de este último. Se evaluará el siguiente ciclo completo de onda, y se seguirá realizando esta operación hasta que un ciclo cumpla la condición y su RMS sea menor que ésta. Entonces estaremos en la siguiente punto.
b) Que el valor de RMS es menor que el valor condición elegido por el usuario, los datos correspondientes a este ciclo evento son añadidos al buffer y se evaluará el ciclo completo de onda siguiente y si éste también cumple la condición se añadirán sus datos al buffer hasta que se encuentre un ciclo que no la cumpla y entonces se añadirá al buffer existente de datos, los datos correspondientes a un periodo de tiempo elegido por el usuario llamado tiempo post-evento. El buffer de datos resultante estará formado por los datos del tiempo pre-evento, los datos del/(de los) ciclo/s evento/s y por los datos del

tiempo post-evento. Luego este buffer será guardado con el nombre que elija el usuario.

En todos estos casos el usuario podrá elegir tanto la frecuencia de muestreo como el nombre del archivo a guardar.

Caso 1º	Caso 2º	Caso 3º
Medir y grabar o permanecer a la espera según lo decida el usuario	**Medir y grabar durante un tiempo y permanecer al espera otro**	**Medir y grabar un gran buffer de datos cuando si cumpla una serie de condiciones**

Desarrollo de la programación: Preliminares

Para la programación de un dispositivo como el CompactRIO, National Instruments establece que se realiza gracias a un "Empty Project" (Proyecto Vacío) que es una de las opciones que hay en la pantalla de inicio al abrir/ejecutar el LabVIEW.

- Ejecutar LabVIEW y en su pantalla de inicio "Getting Started" seleccionar "Empty Project" y se nos abre la siguiente pantalla:

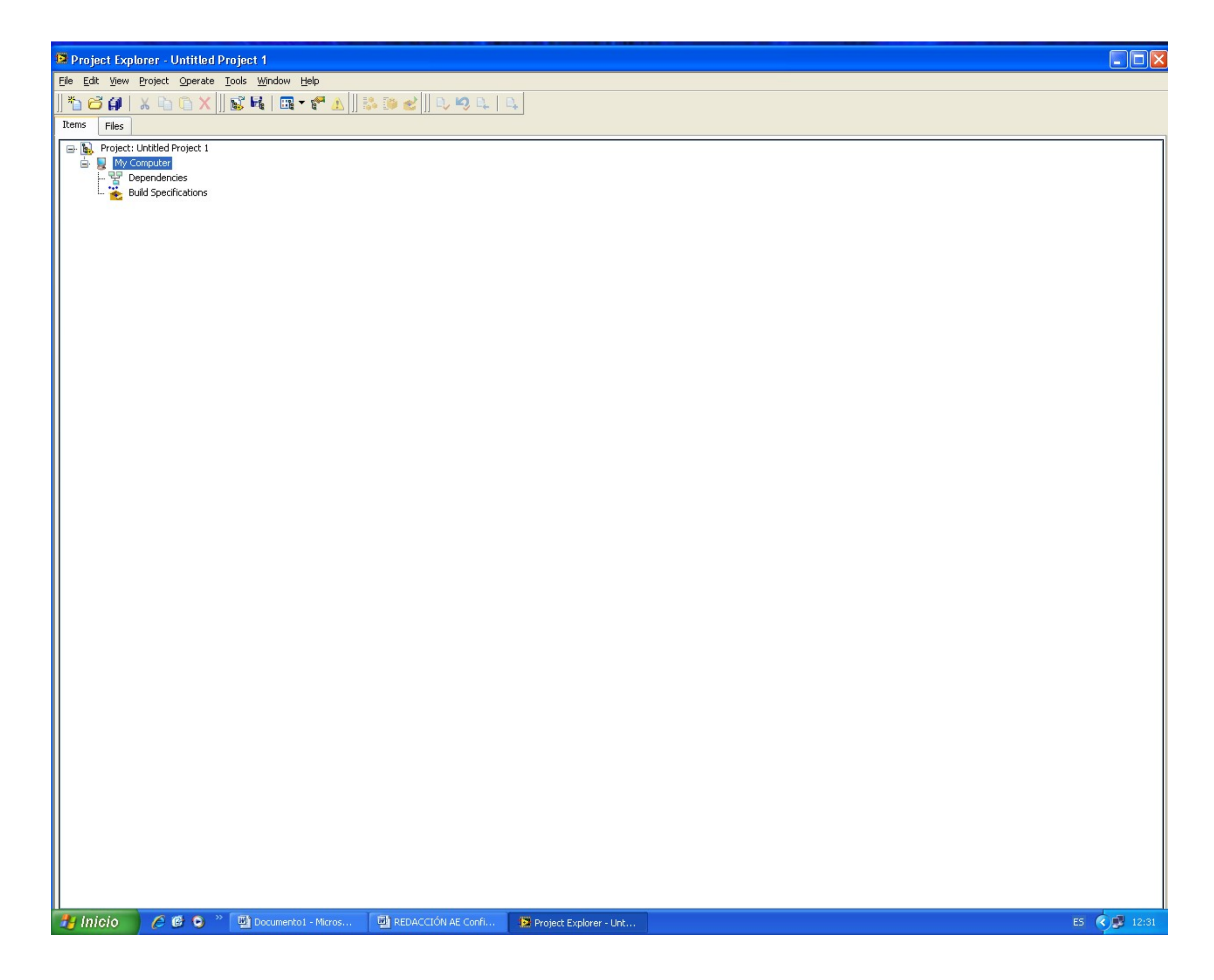


Figura 9: Inicio del proyecto ("Empty Project").

- Clickear con el botón derecho sobre el icono de "Project: Untitled Project 1" y seleccionar "Targets and Devices"

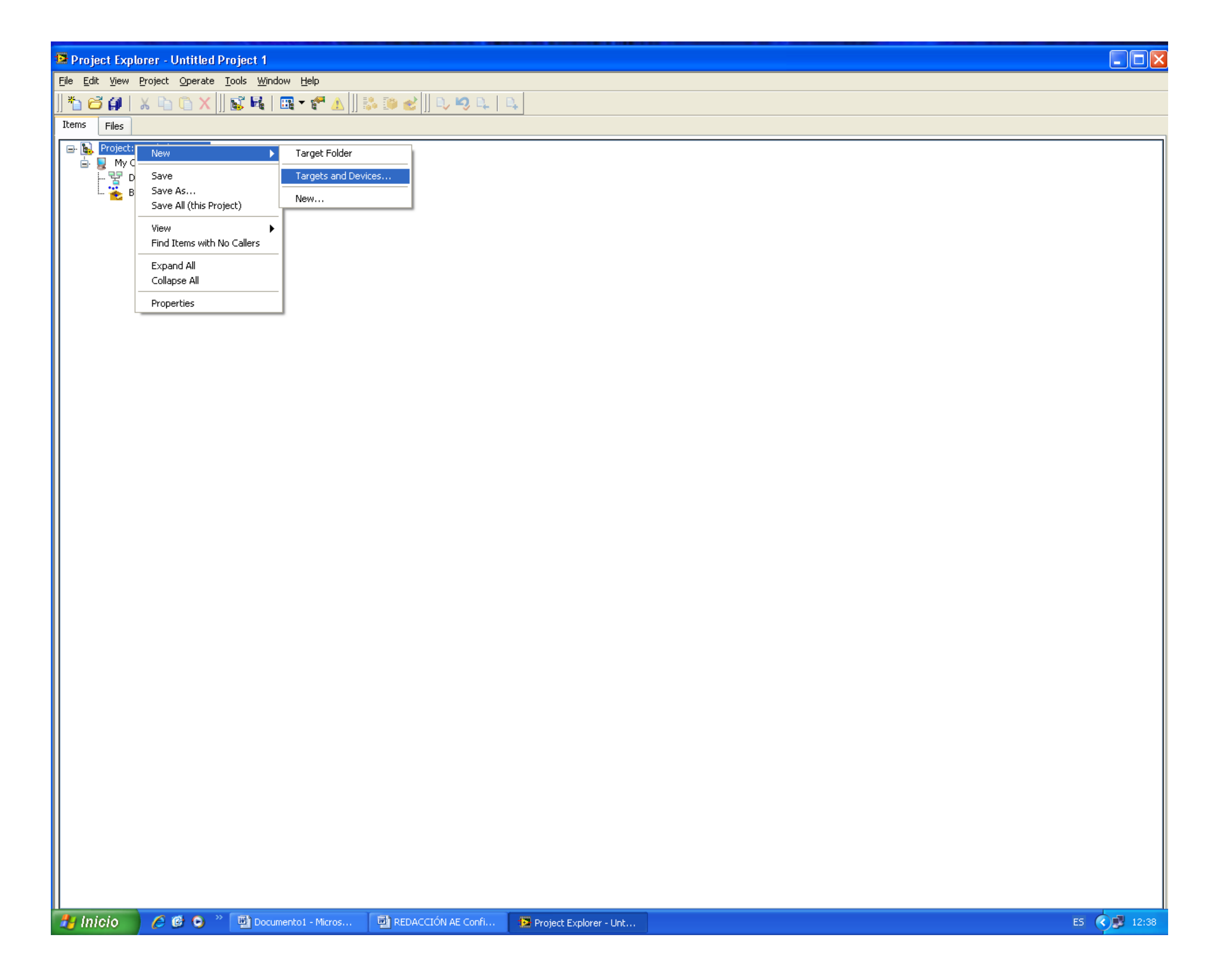


Figura 10: Segundo paso del proyecto ("Empty Project").

- Seleccionar “Real-Time CompactRIO”

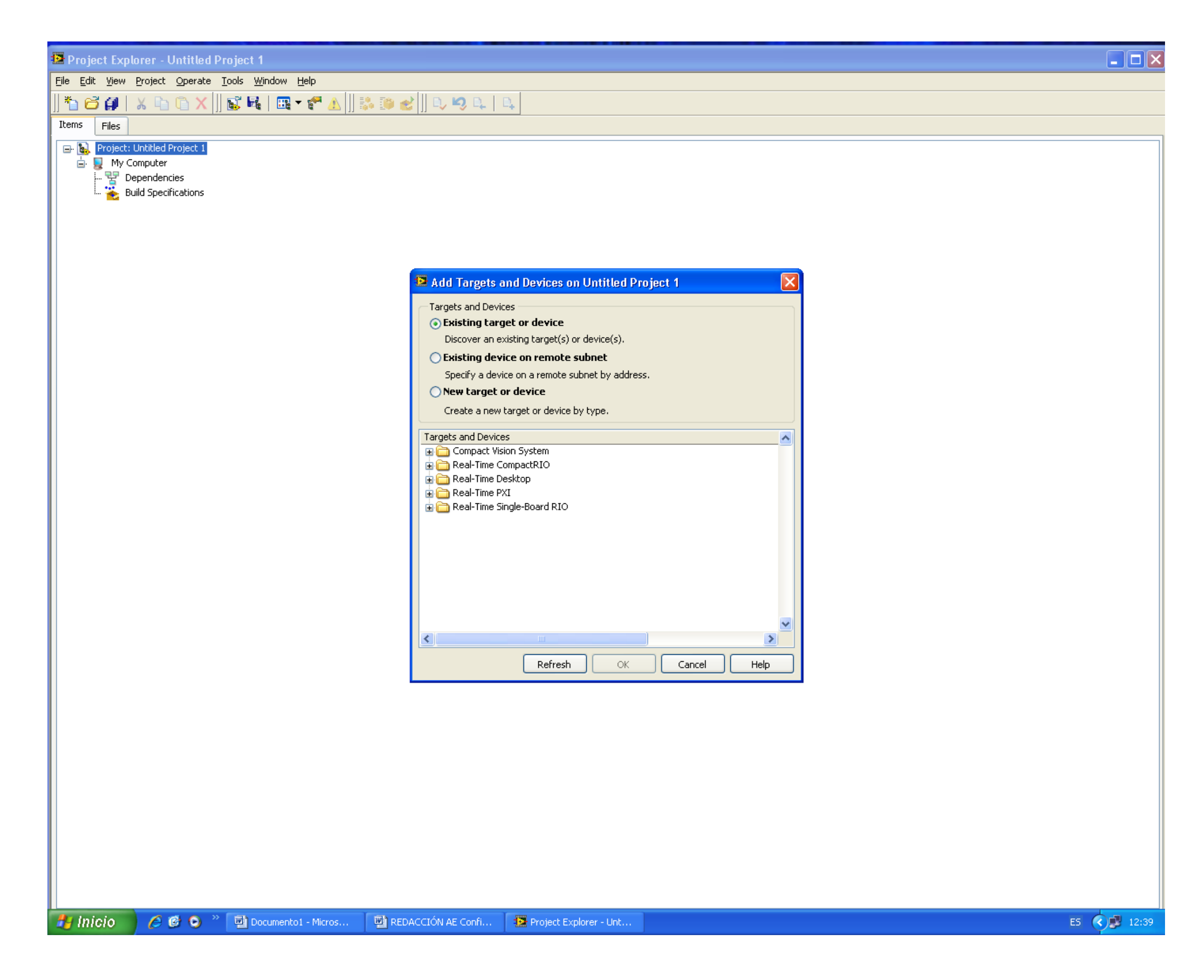


Figura 11: Tercer paso del proyecto ("Empty Project").

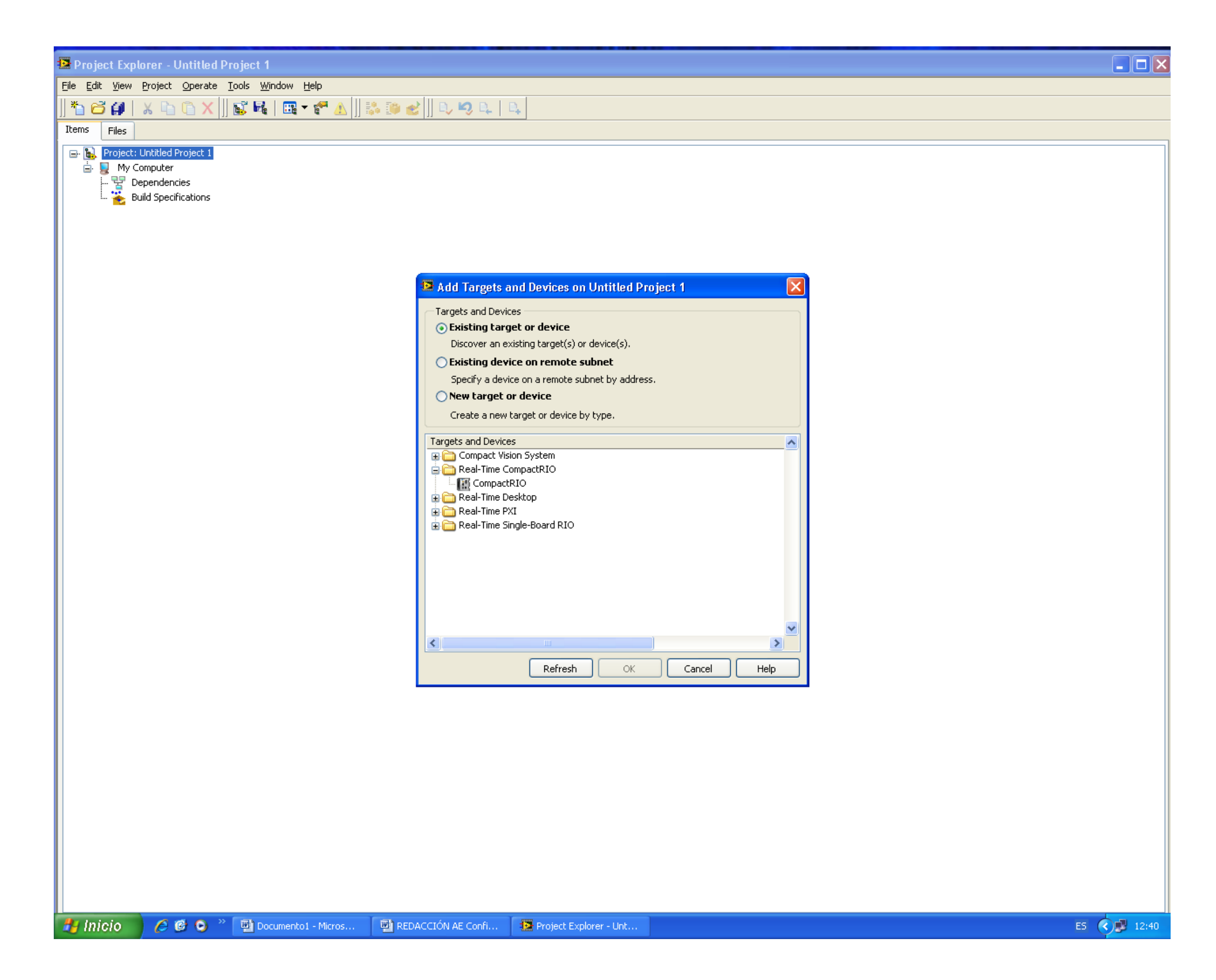


Figura 12: Cuarto paso del proyecto ("Empty Project").

- Seleccionar “CompactRIO” y ya hemos configurado las tarjetas de adquisición obteniendo la siguiente pantalla, donde tenemos todos los canales accesibles para su uso. Podemos así, integrar estos canales de las tarjetas en cualquier programa o proyecto de LabVIEW.

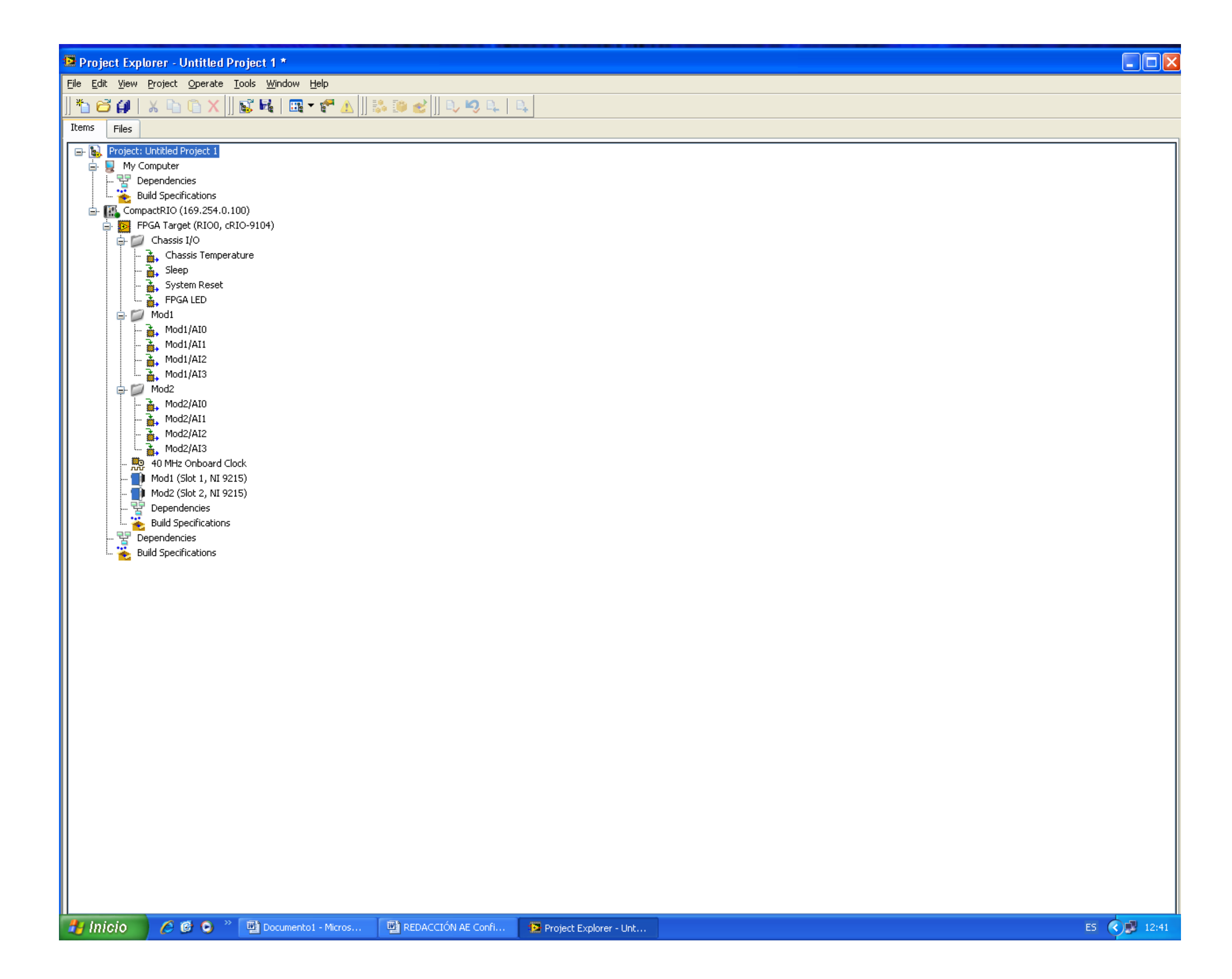


Figura 13: Proyecto ("Empty Project") listo para progrmar los VI´s ("Virtual Instrument").

Introducción de archivos Virtual Instruments (VI)

Ahora hay que cargar en el "Empty Project" dos archivos .vi, uno para programar las tarjetas de adquisición de datos y otro para programar el CompactRIO.

VI de las tarjetas de adquisición de datos

Para cargar un archivo .vi en el ámbito de las tarjetas de adquisición de datos hay que clickear con botón derecho sobre "FPGA Target" y seleccionar dentro de "New" la opción "VI", como se puede ver en la Figura 14.

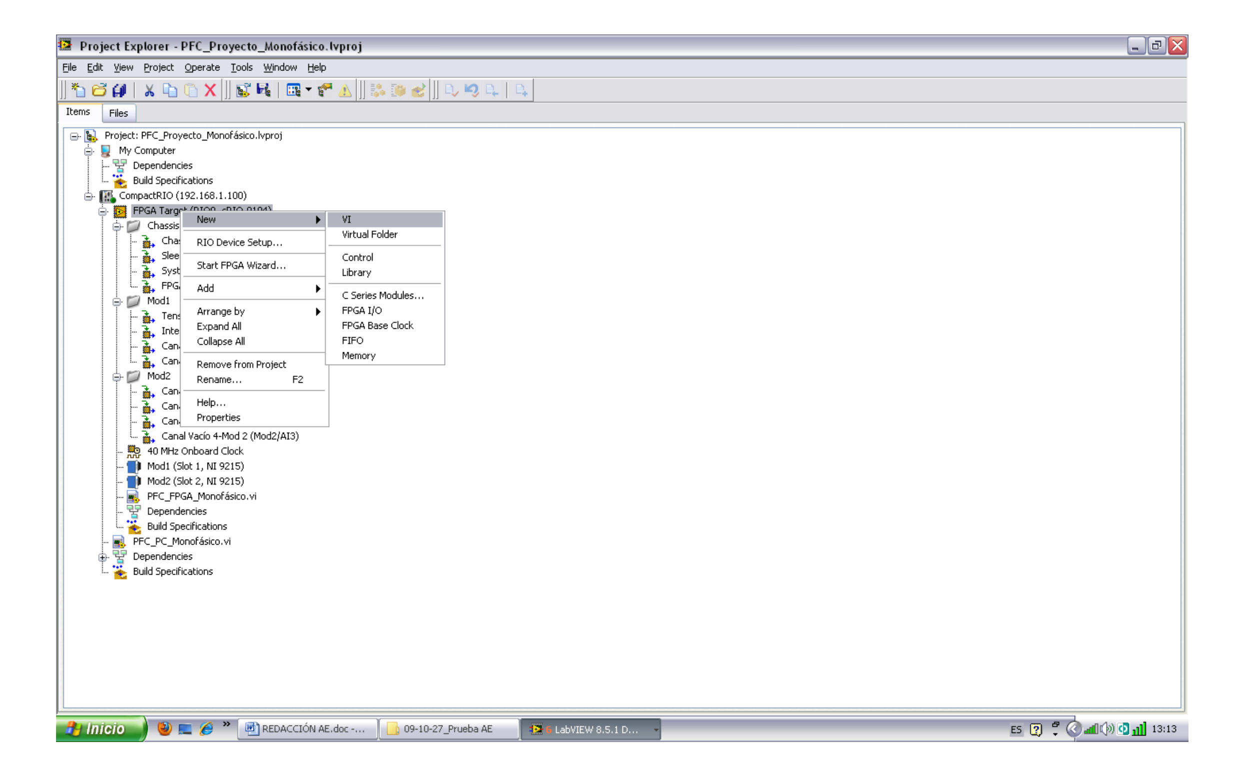


Figura 14: Cargar un archivo .vi en el ámbito de las tarjetas de adquisición de datos.

En nuestro proyecto queremos que las tarjetas tengan abiertos una serie de canales como entradas de datos, es decir, que estén estos canales midiendo/leyendo en continuo.

Para el caso del sistema monofásico sólo necesitaremos una tarjeta, y por ejemplo, los dos primeros canales, el primero para la tensión y el segundo para la intensidad. Tal y como se puede ver en la Figura 15.



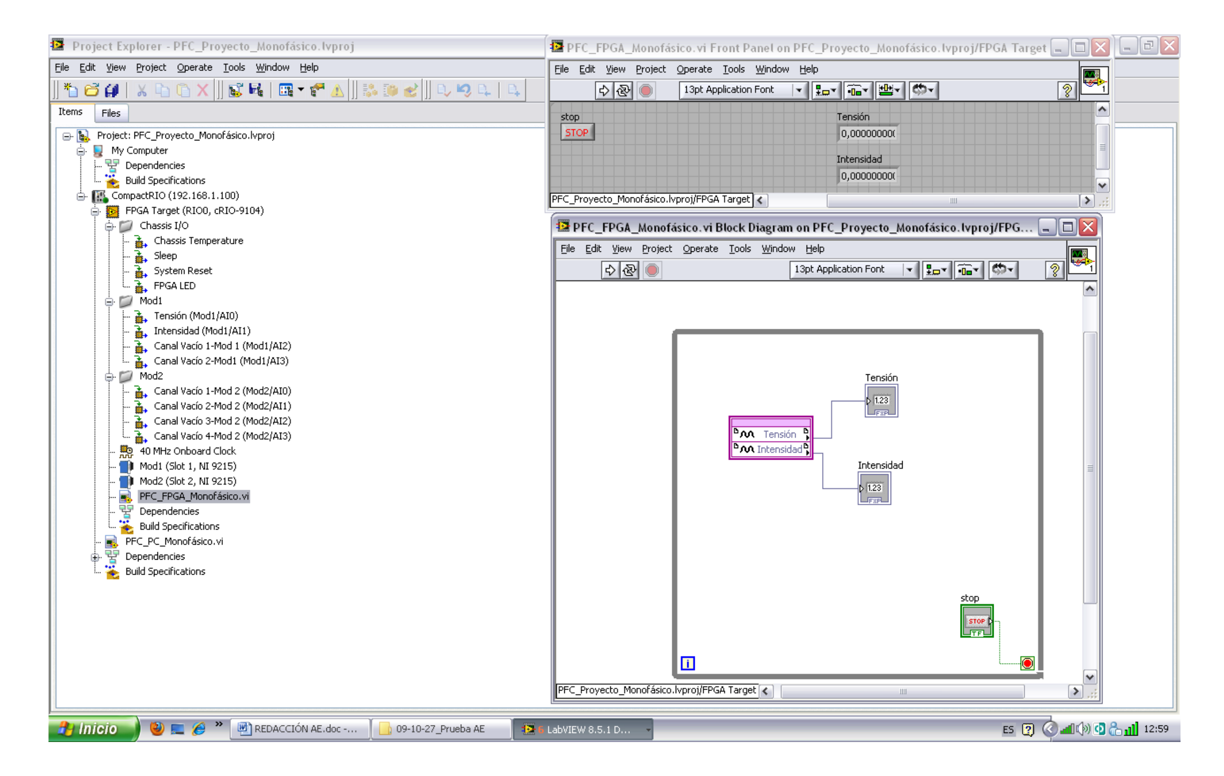


Figura 15: Sistema Monofásico_FPGA.

Para el caso de un sistema trifásico será necesarias las dos tarjetas y en este caso utilizaremos los tres primeros canales de la primera tarjeta para la tensiones y los tres primeros de la segunda tarjeta para la intensidades tal y como se puede apreciar en la Figura 16.

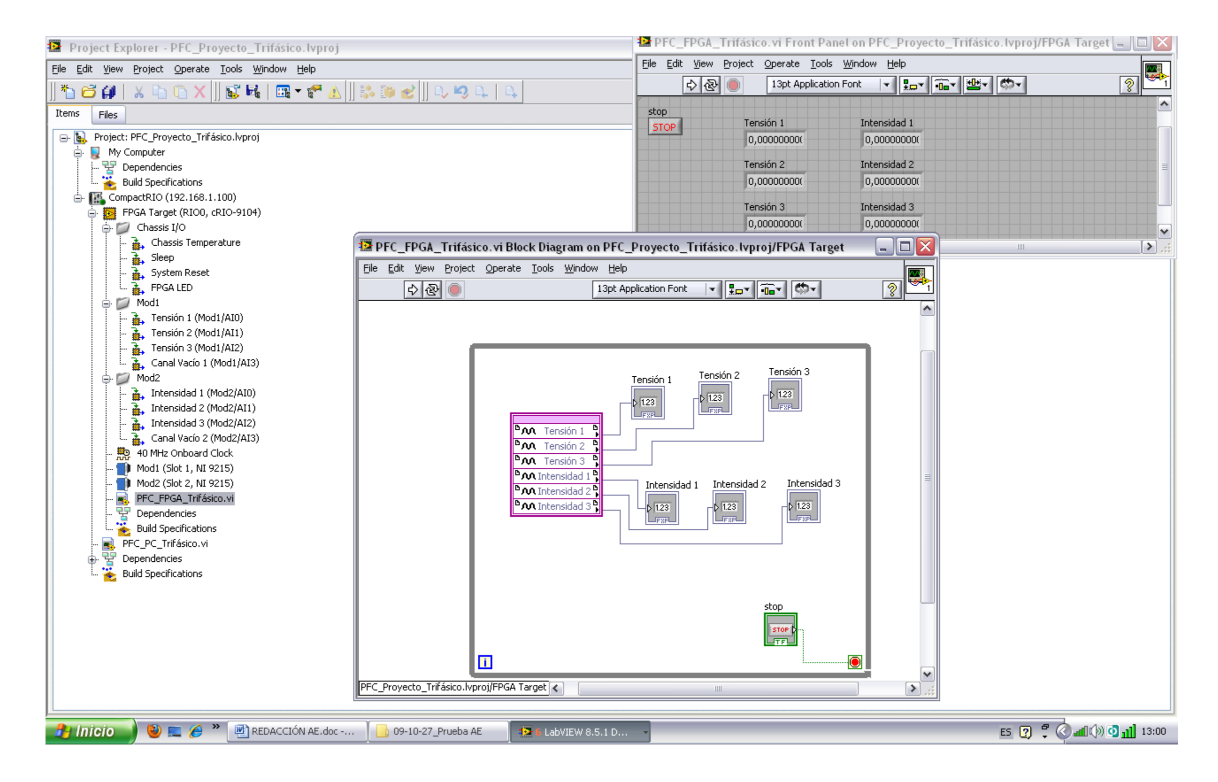


Figura 16: Sistema Trifásico_FPGA.

VI del CompactRIO

Para cargar un archivo .vi en el ámbito del CompactRIO, para que éste realice la tareas pretendidas, hay que clickear con botón derecho sobre "CompactRIO (192.168.1.100)" y seleccionar dentro de "New" la opción "VI", como se puede ver en la Figura 17.

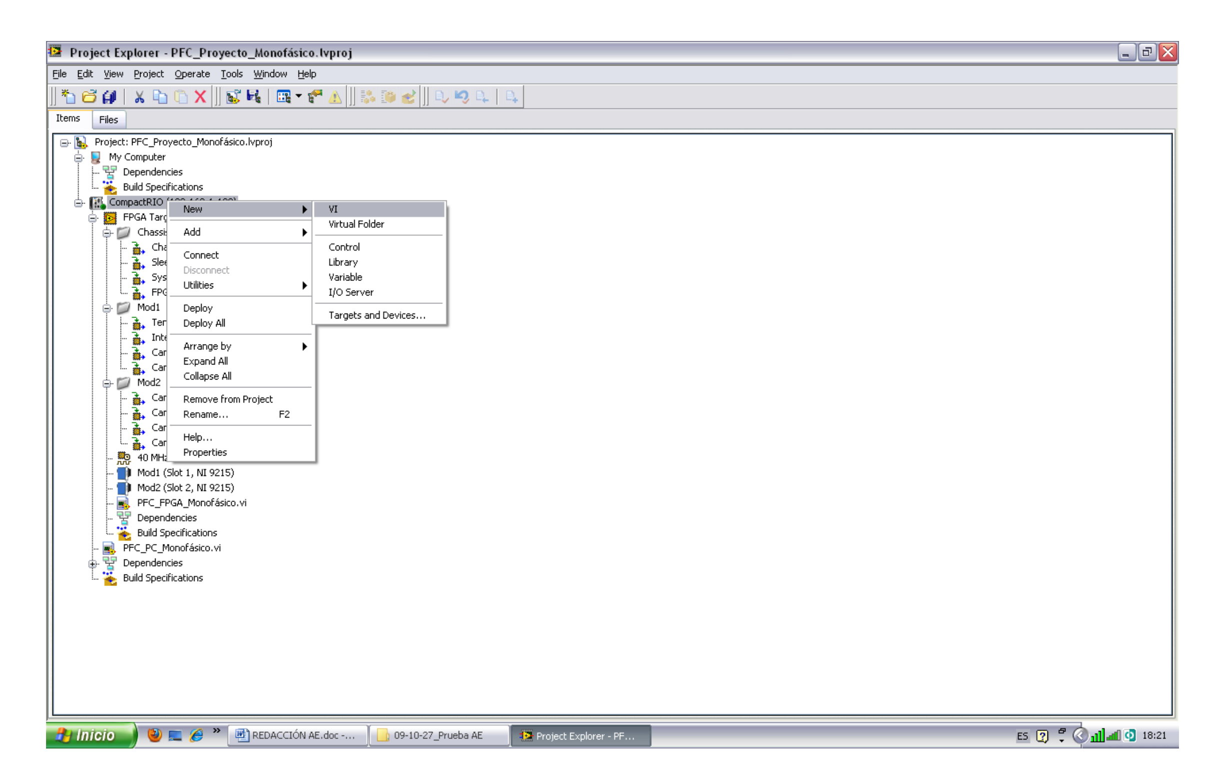


Figura 17: Cargar un archivo .vi en el CompactRIO.

Programación de los archivos Virtual Instruments (VI)

VI de las tarjetas de adquisición de datos

Para la programación de las tarjetas utilizamos la estructura "While Loop", tal y como se puede apreciar tanto en la Figura 15: Sistema Monofásico_FPGA. como en la Figura 16: Sistema Trifásico_FPGA., que se puede seleccionar en la "Functions Palette / Programmimg / Structures". Con esta estructura se está realizando de forma cíclica la programación que hay en su interior hasta que se cumpla la condición de parada que en

nuestro caso es un "Control" (así son llamadas las entradas en LabVIEW) booleano de "Stop". Para abrir los canales de las tarjetas "pinchamos y arrastramos" con el "ratón" cualquier canal, que vayamos a utilizar, de la pantalla del "Project Explorer" a la pantalla del "Block Diagram" del VI de las tarjetas de adquisición de datos. Posteriormente agrandaremos esta caja hasta conseguir el número de canales que queramos abrir. Cada tarjeta tiene cuatro canales, por lo que para el sistema monofásico necesitaremos dos canales de la primera tarjeta y para el sistema trifásico tres de cada tarjeta. Luego para seleccionar cada canal en la posición de la caja, basta con clickear con botón derecho y seleccionar el canal que queramos, como se puede ver en las siguientes figuras.

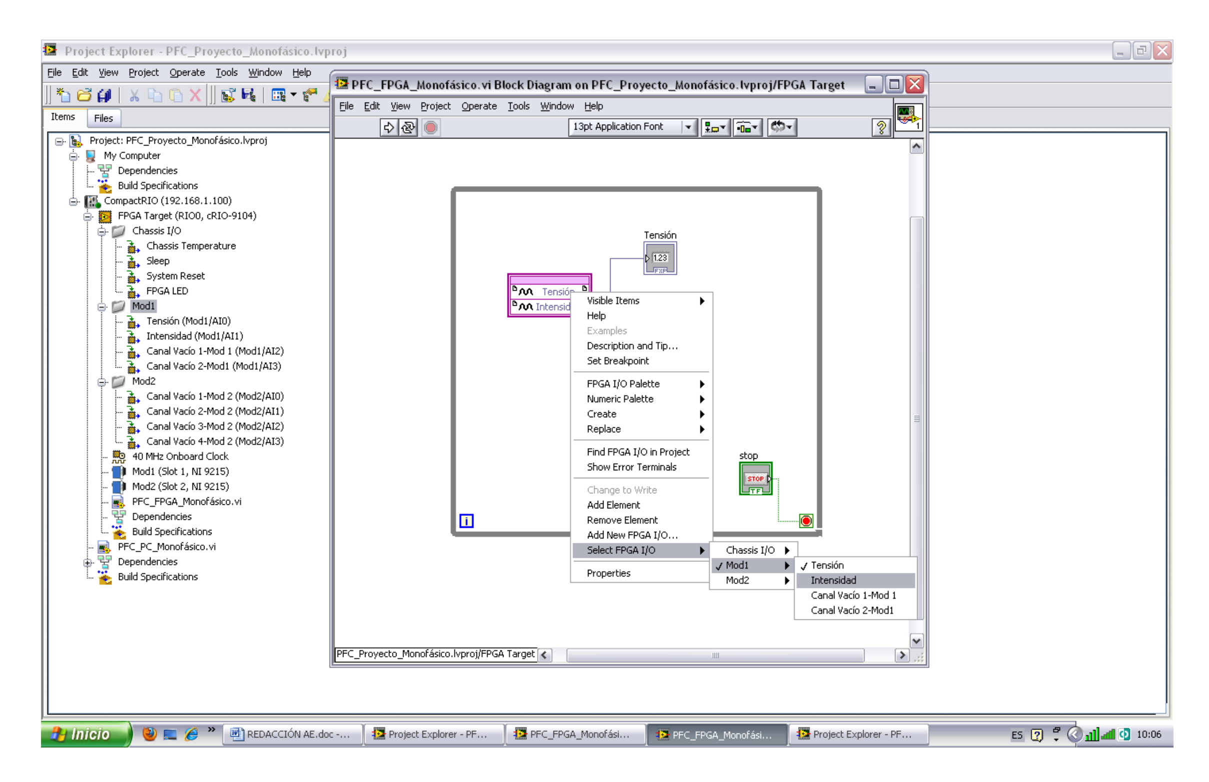


Figura 18: Selección de canales_Sistema Monofásico.

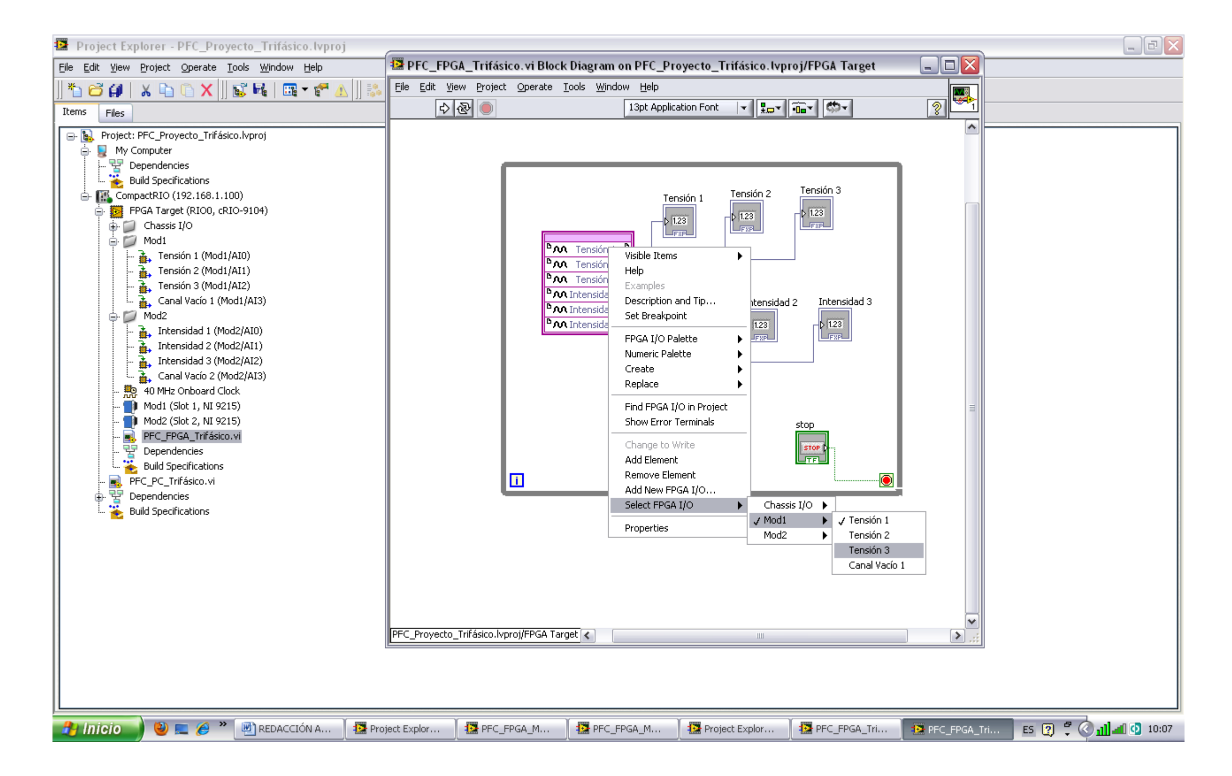


Figura 19: Selección de canales_Sistema Trifásico.

También para poder observar las medidas de los canales hemos introducido en cada canal un "Indicador" (así son llamadas las salidas en LabVIEW). Estas medidas se pueden ver en el "Front Panel" como se puede ver en la Figura 15: Sistema Monofásico_FPGA. y en la Figura 16: Sistema Trifásico_FPGA..

Para que el VI de las tarjetas de adquisición de datos pueda ser llamado por el VI del CompactRIO éste debe ser compilado dentro del "Project Explorer". Para realizar la compilación basta con clickear con botón derecho sobre el nombre del VI de las tarjetas de adquisición de datos y selleccionar la opción de compilar, tal y como se puede ver en la Figura 20.

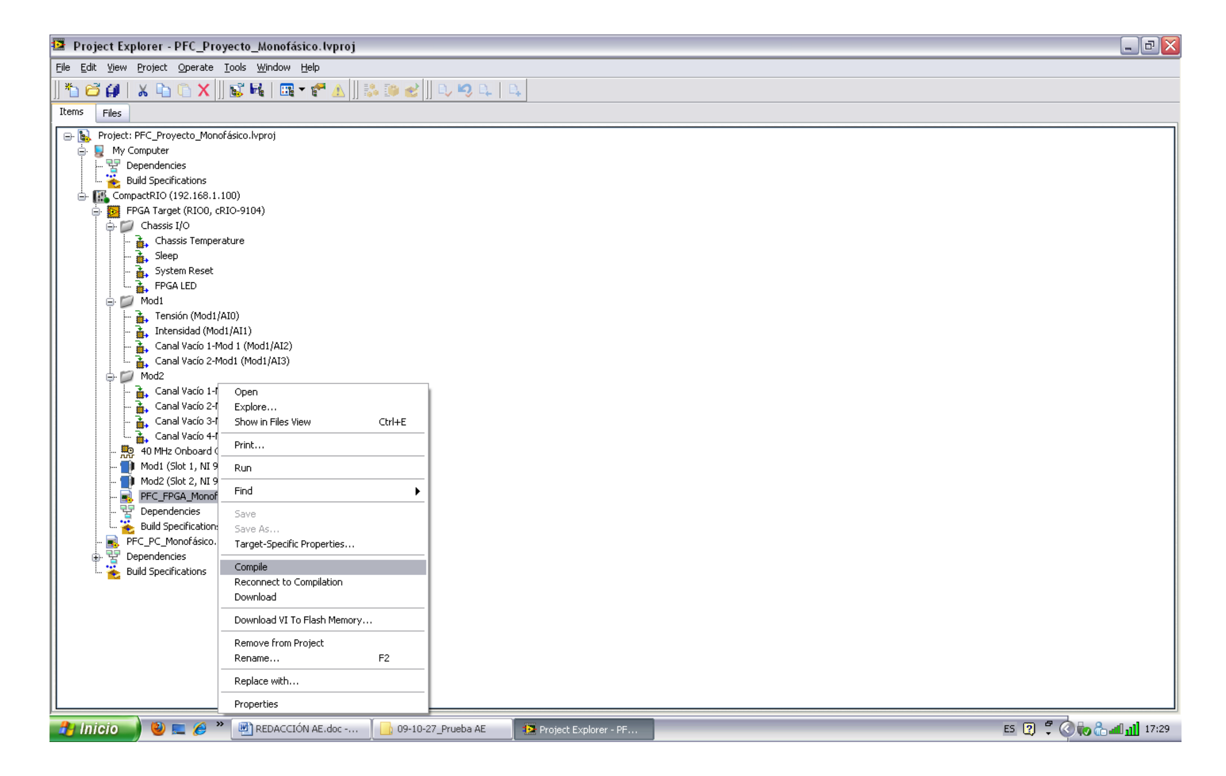


Figura 20: Compilar el VI de las tarjetas de adquisición de datos.

VI del CompactRIO

Para cada uno de los tres casos explicados en el apartado . Introducción y requerimientos hemos utilizado una estructura "While Loop" para poder elegir una frecuencia de muestreo, así como poder parar la ejecución de cada caso de forma independiente.

Esta frecuencia de muestreo se consigue gracias a la función "Wait Until Next ms Multiple" que se encuentra en la paleta de funciones en "Programming/Timing" y así se puede determinar el tiempo que trascurre entre cada iteración de la estructura "While Loop".

También tenemos que abrir o llamar al VI de las tarjetas de adquisición, siempre y cuando haya sido compilado previamente, con la función "Open FPGA VI Reference (FPGA Interface)" que se encuentra en la paleta de funciones en "FPGA Interface". Véase en las figuras: Figura 21 y Figura 22.

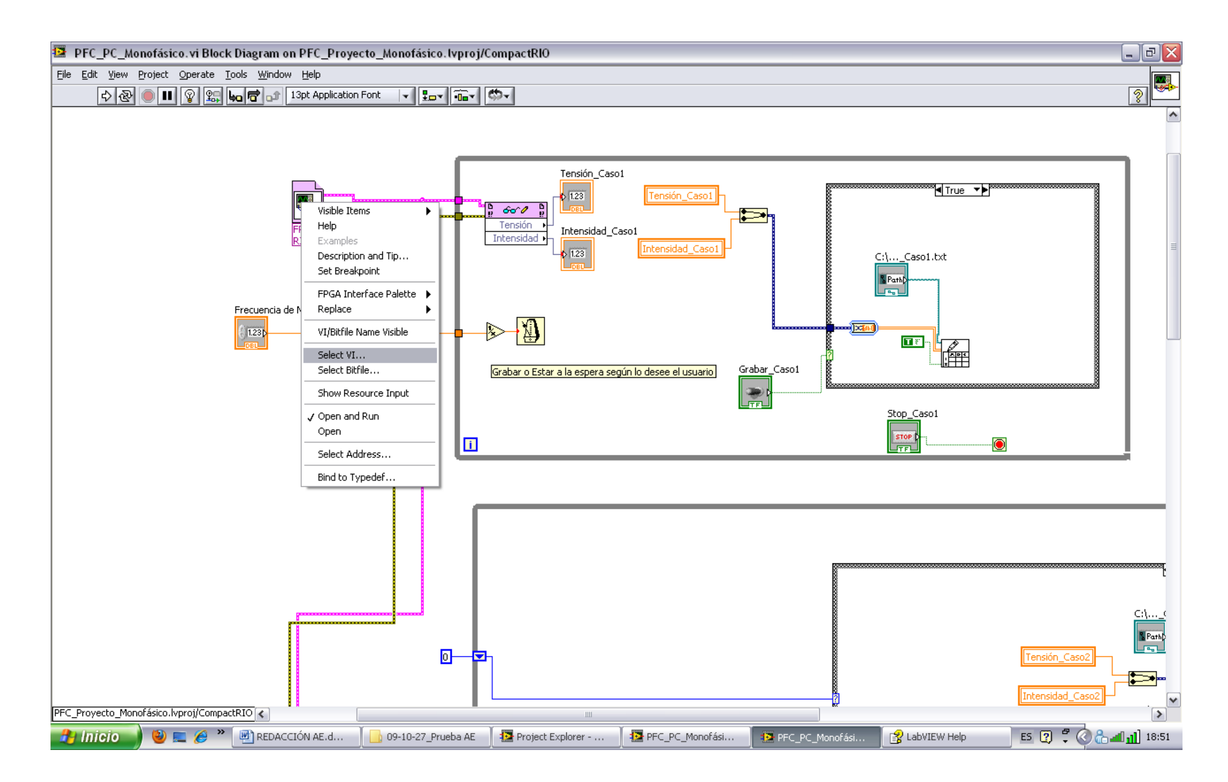


Figura 21: Seleccionar el VI en la función Open FPGA VI Reference (FPGA Interface)_1.

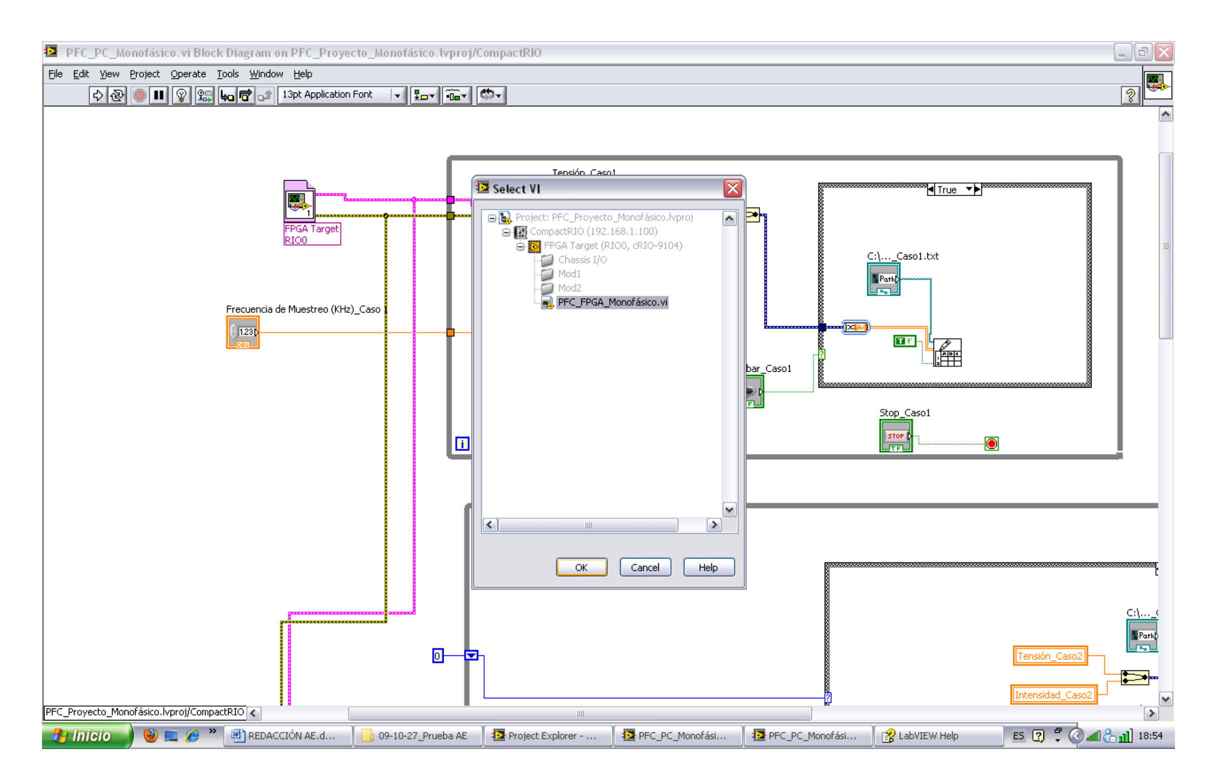


Figura 22: Seleccionar el VI en la función Open FPGA VI Reference (FPGA Interface)_2.

En cada uno de las tres estructuras será necesario utilizar la función "Read/Write Control (FPGA Interface)" (Paleta de funciones/FPGA Interface)

para poder abrir y leer los canales de las tarjetas de adquisición de datos y por lo tanto tratarlos.

Todo este esquema de procedimiento, estructuras y funciones han sido utilizadas de forma semejante tanto para el sistema monofásico y como para el sistema trifásico.

Grabar los datos medidos o permanecer a la espera según lo estime oportuno el usuario

En este caso sólo tenemos dos tareas a realizar grabar o permanecer a la espera por lo tanto lo más sencillo es utilizar una estructura "Case Structure" (Paleta de funciones/Structures) dónde nuestro caso falso será permanecer a la espera y el verdadero será grabar un .txt con el nombre que elija el usuario en el "Panel Frontal". La condición de verdadero o falso se establece con un "Control" (entrada en LabVIEW) de tipo "booleano normalmente falso", que el usuario podrá accionar en el "Panel Frontal" a verdadero o falso; así como el "booleano de stop" cuando lo estime oportuno. La programación se puede observar en las siguientes figuras: Figura 23, Figura 24, Figura 25 y Figura 26.

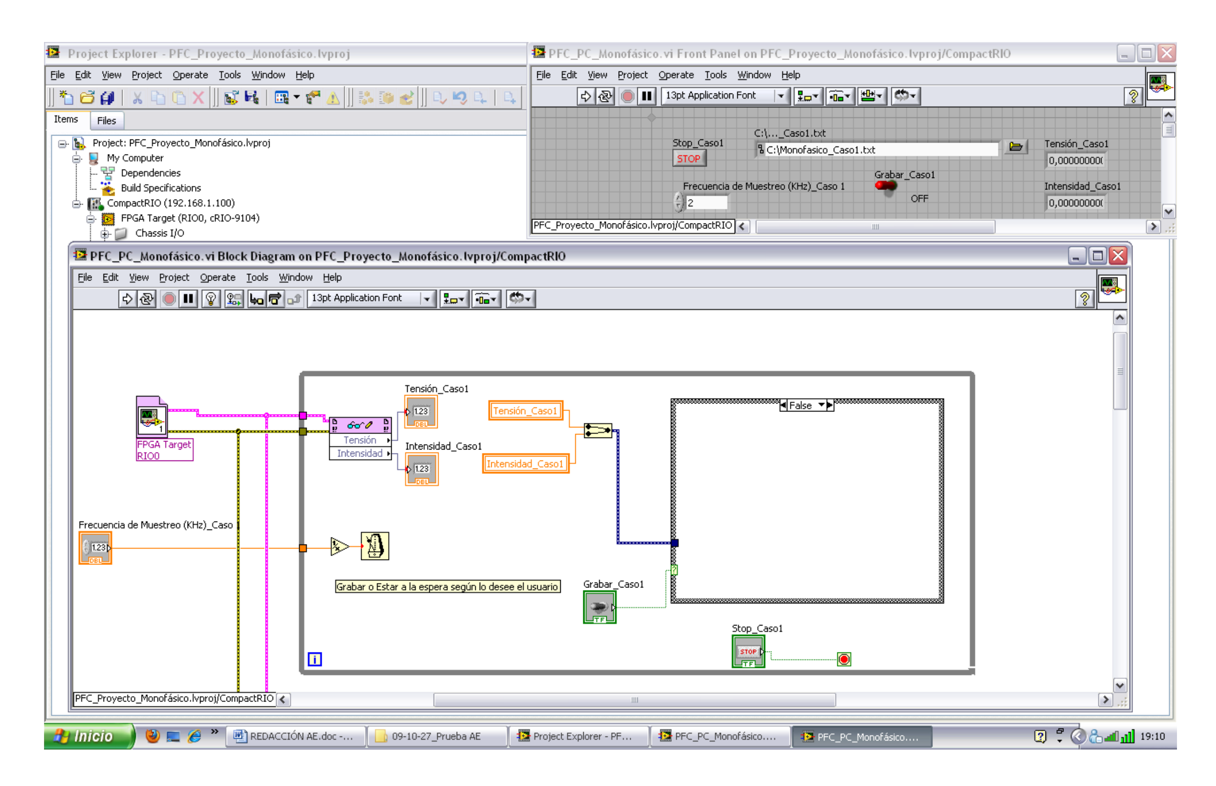


Figura 23: VI del primer caso_grabar OFF_Monofásico.

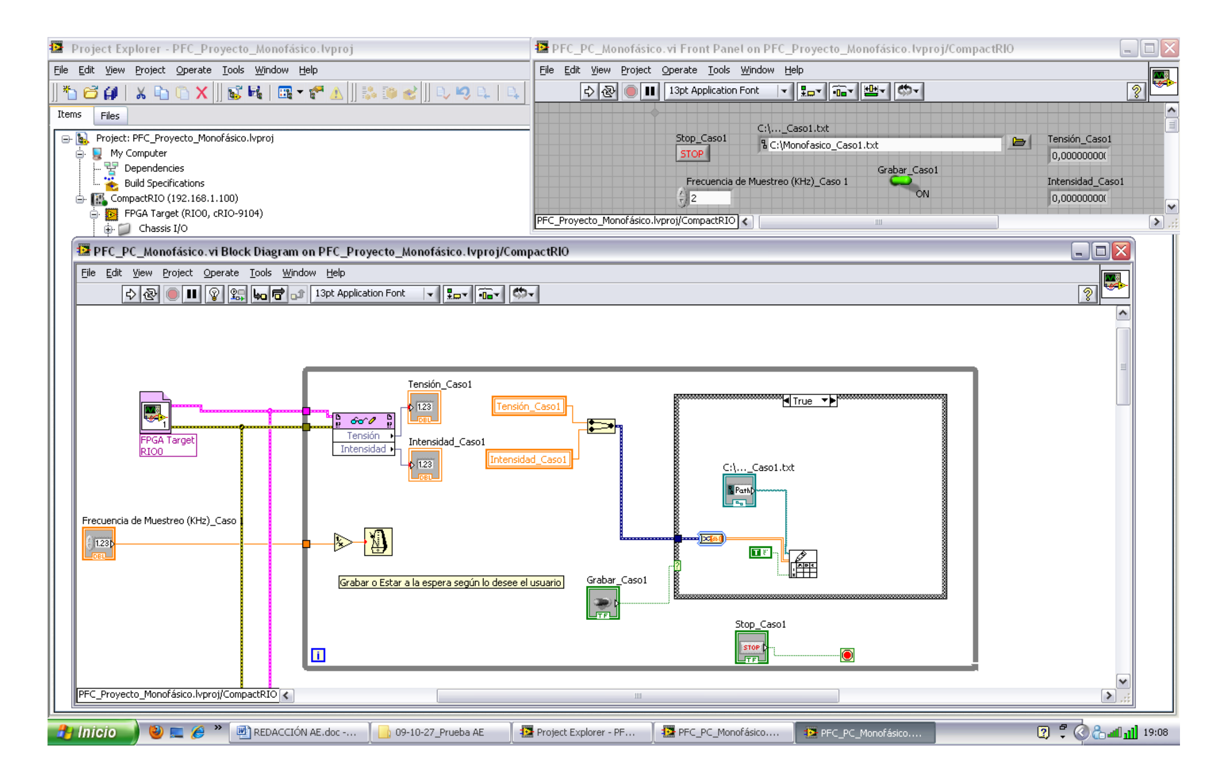


Figura 24: VI del primer caso_grabar ON_Monofásico.

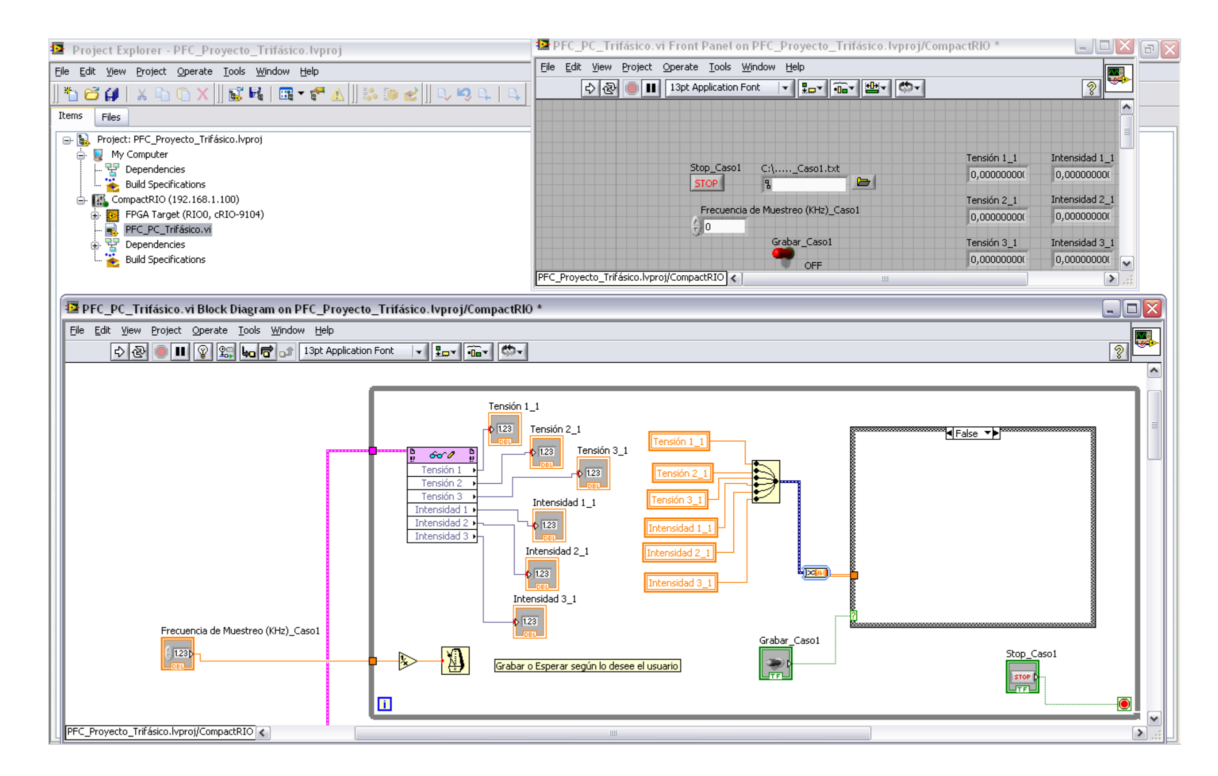


Figura 25: VI del primer caso_grabar OFF_Trifásico.

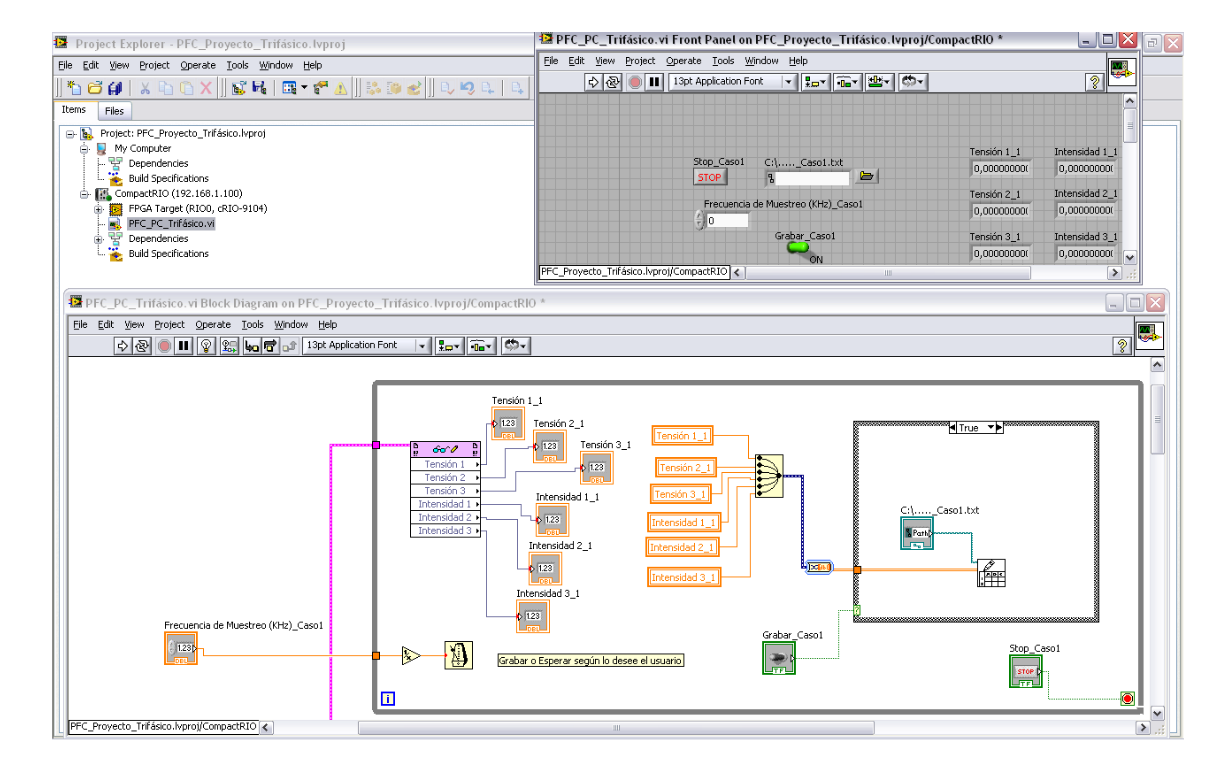


Figura 26: VI del primer caso_grabar ON_Trifásico.

En los canales abiertos con la función "Read/Write Control (FPGA Interface)" se creará en cada uno de ellos un "Indicador" (salida en LabVIEW) a su vez con estos unas "Variables Locales". La precisión de estos indicadores las ampliaremos a "Doble Precision". Tal y como se puede ver en las siguientes figuras: Figura 27, Figura 28 y Figura 29 (se realiza de igual manera en el sistema trifásico).

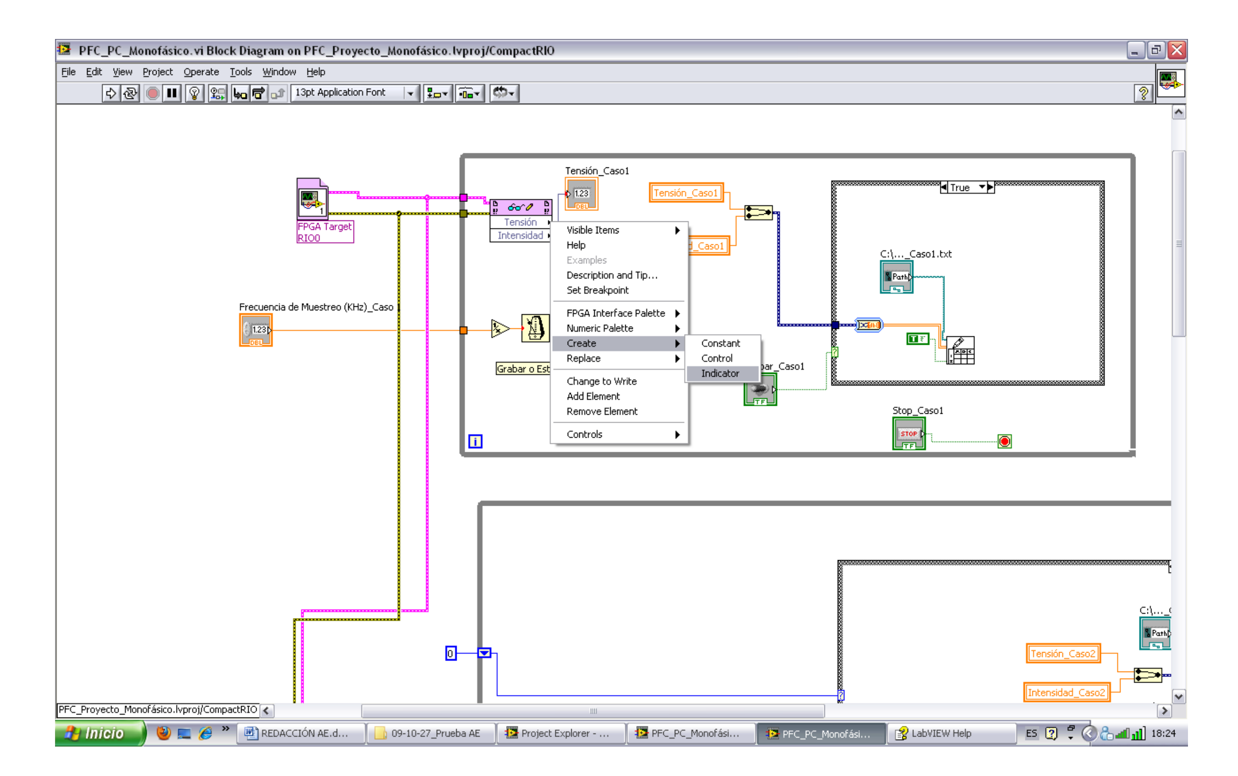


Figura 27: Crear un Indicador en los canales de lectura.

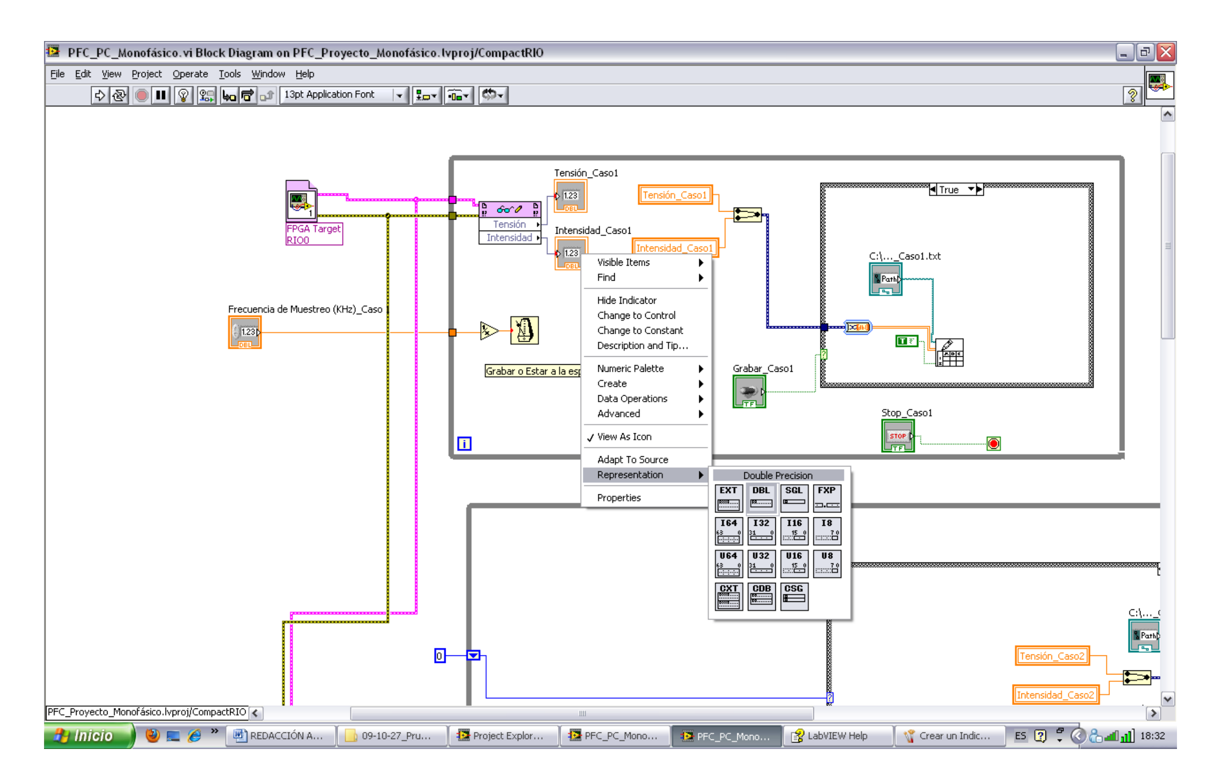


Figura 28: Cambiar la precisión de representación de un indicador.

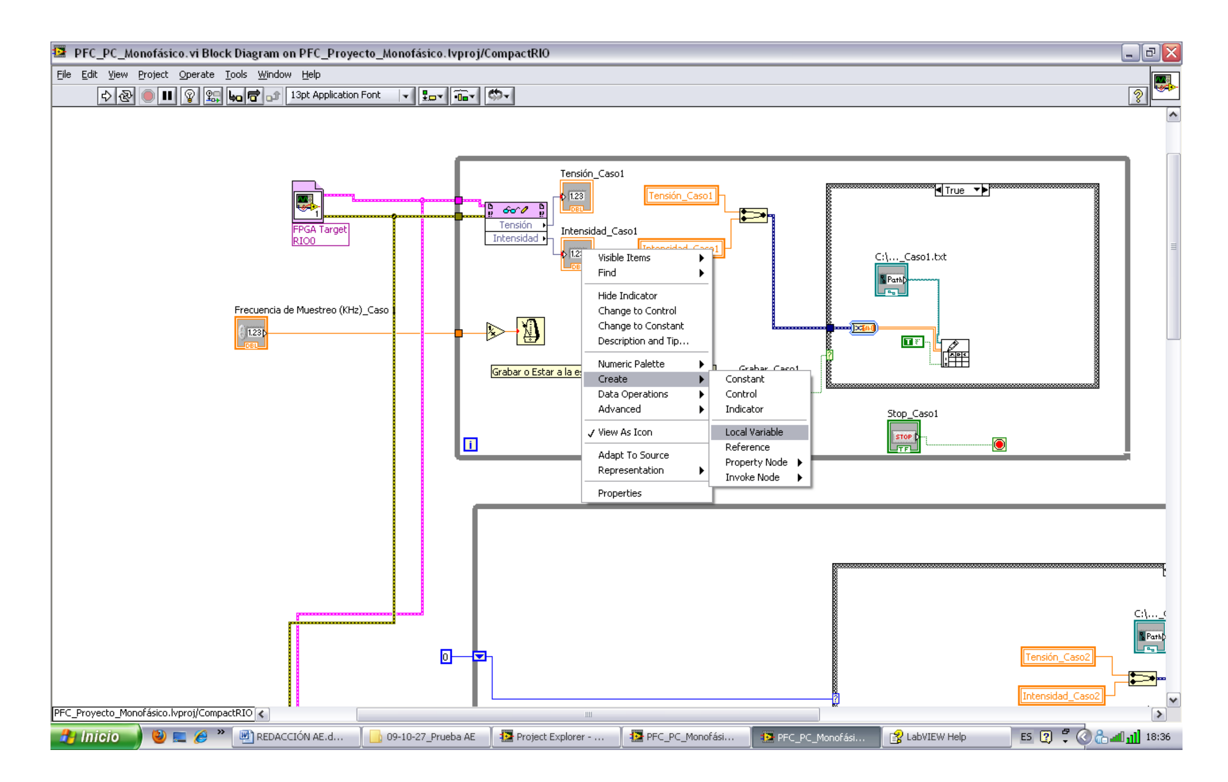


Figura 29: Crear una "Local Variable".

Los datos recogidos en los canales son unidos por la función “Merge Signals” (Paleta de funciones/Express/Signal Manipulation) y convertidos en un array dinámico de datos con la función “Convert from Dynamic Data” (Paleta de funciones/Express/Signal Manipulation) que hay que configurarla de tal manera que cada columna sea para un canal por lo tanto: (Figura 30)

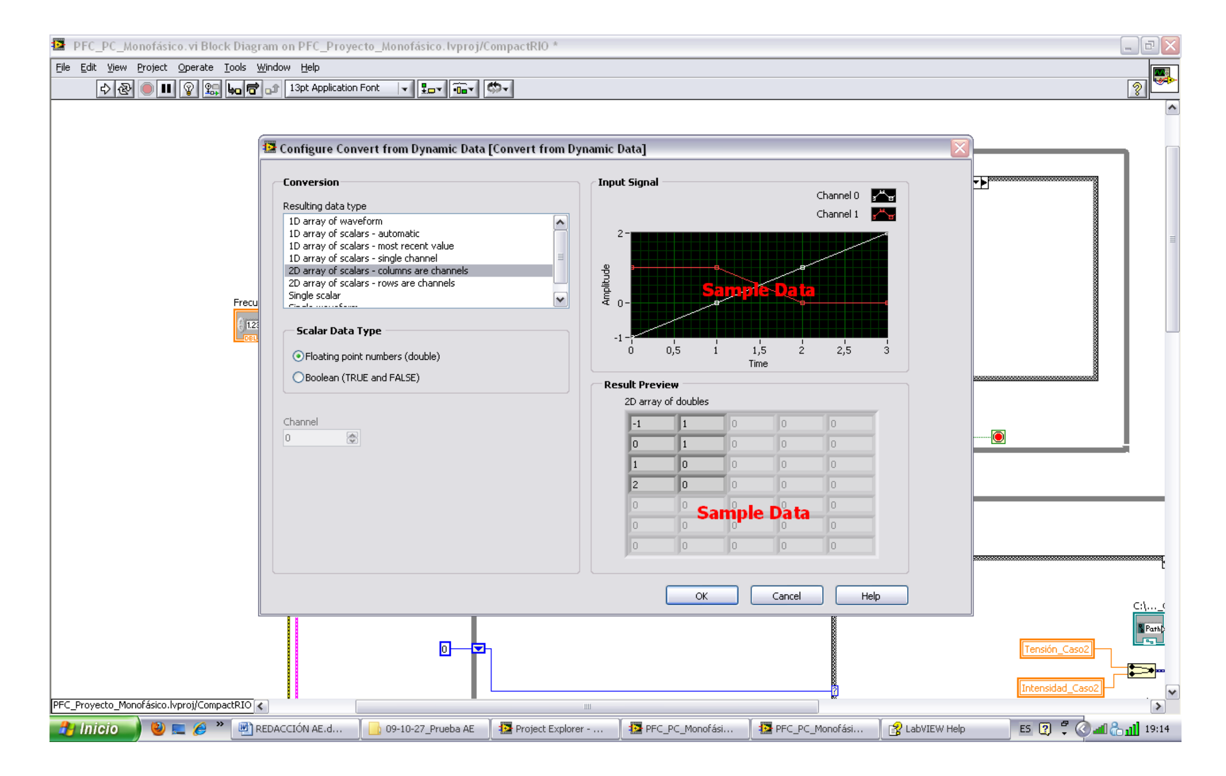


Figura 30: Configuración de la función "Convert from Dynamic Data".

Para la grabación utilizamos la función “Write To Spreadsheet File” (Paleta de funciones/Programming/File I/O) donde en el terminal “If **append to file?**” colocamos el “booleano true” para que el último dato al entrar no machaque al anterior sino que lo incorpore en la última posición de un listado de datos. En el terminal “**file path**” conectamos la función “Path Constant” (Paleta de funciones/Programming/File I/O/File Constants) y posteriormente lo convertimos en “Control” para que el usuario en panel frontal pueda elegir su nombre (véase Figura 31).

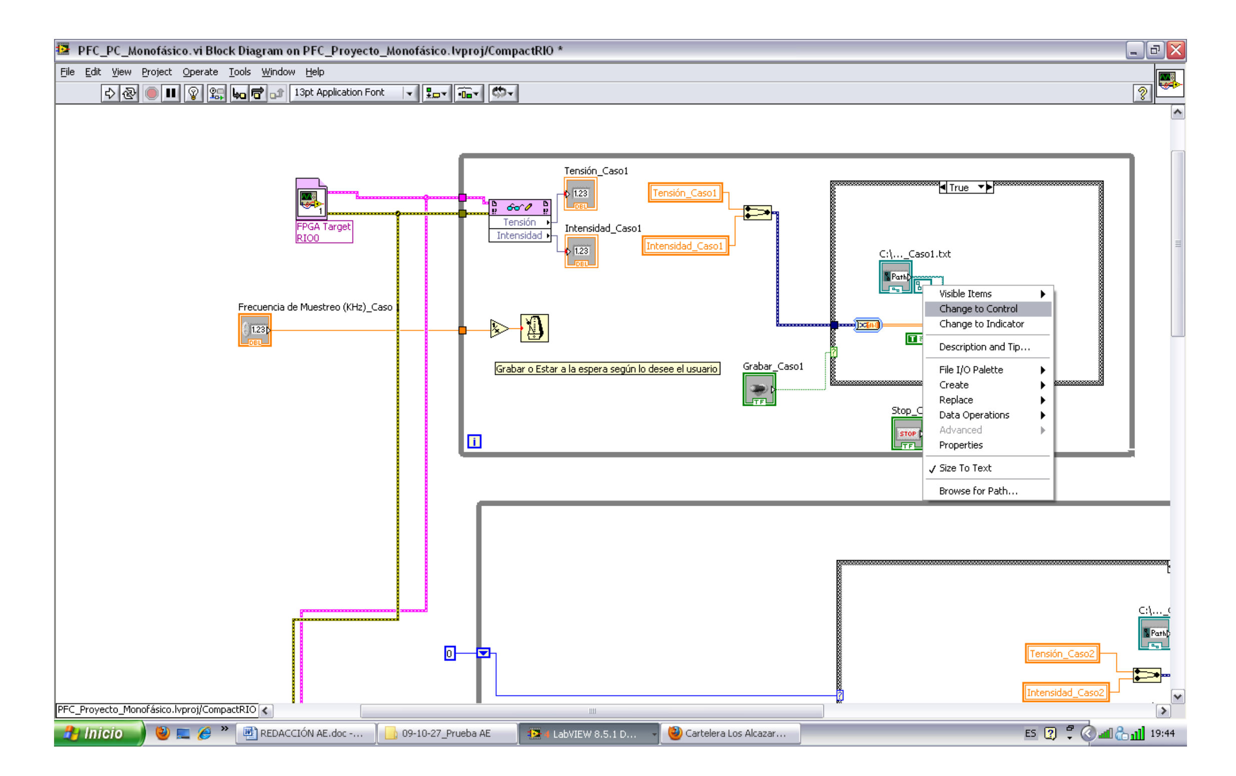


Figura 31: Cambiar la función "Path Constant" a un Path Control.

Grabar los datos durante un tiempo y permanecer a la espera durante otro periodo de tiempo

En este caso el usuario en el Panel Frontal podrá elegir la frecuencia de muestreo, el nombre del archivo a guardar (por lo tanto su ubicación, es decir, comenzando el nombre con C:\... decimos que el archivo sea guardado en el disco duro del CompactRIO y con la terminación .txt el formato de éste. Nota: no utilizar tildes en los nombres), el tiempo de muestreo que es el de grabación y el tiempo en espera. En la siguiente figura (Figura 32 y Figura 35) se pude observar estos “Controles” tanto en el Panel Frontal como en el Diagrama de Bloques.

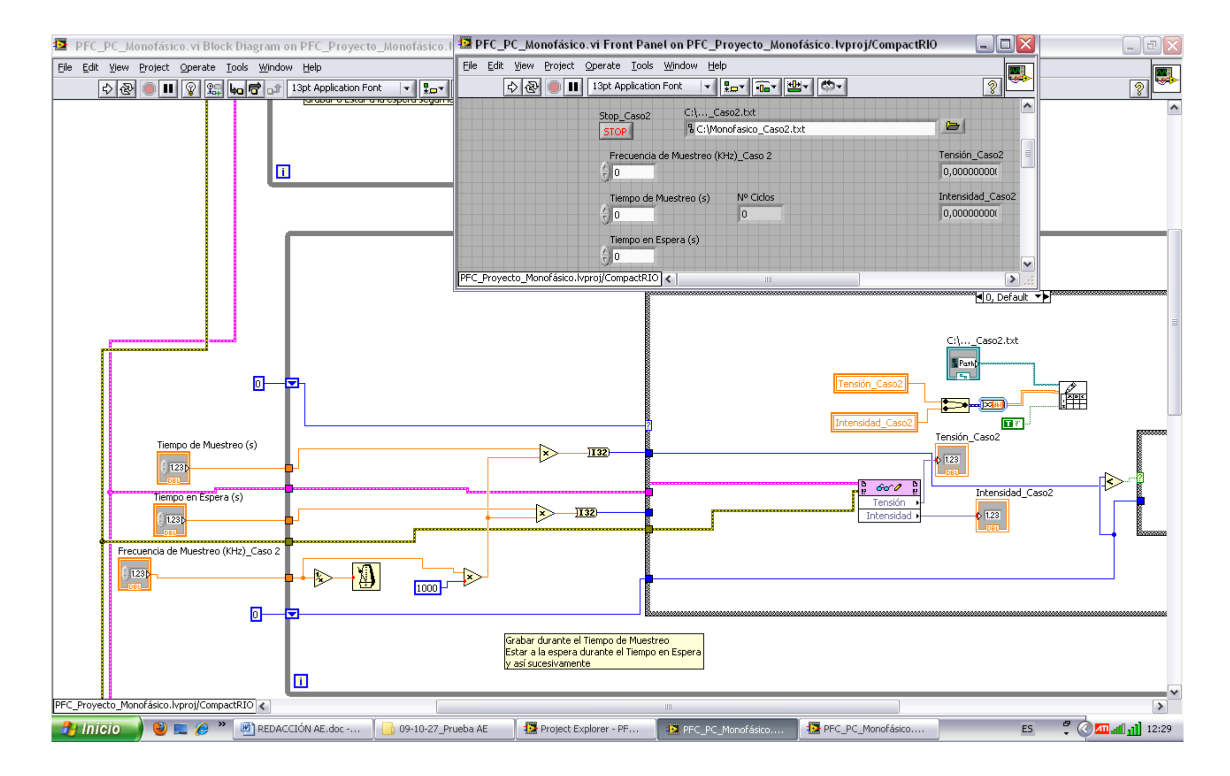
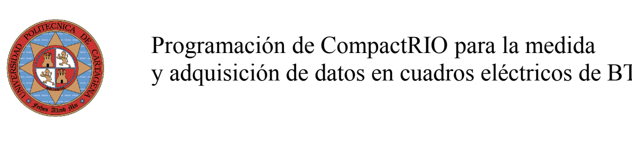


Figura 32: VI del segundo caso_vista general_Monofásico.

Para conseguir que esté primero un tiempo grabando y posteriormente otro en espera utilizaremos una estructura "Case Structure" pero utilizada de una forma diferente a la utilizada en el caso anterior (.Grabar los datos medidos o permanecer a la espera según lo estime oportuno el usuario, Figura 23 y Figura 24). Ahora en el terminal "Case Selector" de la estructura "Case Structure" no introduciremos una condición booleana de verdadero o falso, sino un contador, donde lo iniciamos en la posición cero y tenemos dos casos, caso 0 y caso 1.

Como se puede ver en las figuras: Figura 32, Figura 33, Figura 34 Figura 35 y Figura 36 se permanecerá en el caso 0, que es el caso dónde se graba los datos, hasta que se cumpla la condición para pasar a permanecer en espera. Esta condición la establece el usuario con un tiempo de muestreo y como también estable la frecuencia de muestreo sabemos cuantos datos corresponde a esos dos parámetros, que es simplemente su multiplicación. Por lo tanto se pasará al caso 1 cuando se grabe ese número de datos y se permanecerá en el caso cero hasta que no se llegue a ese número de datos.



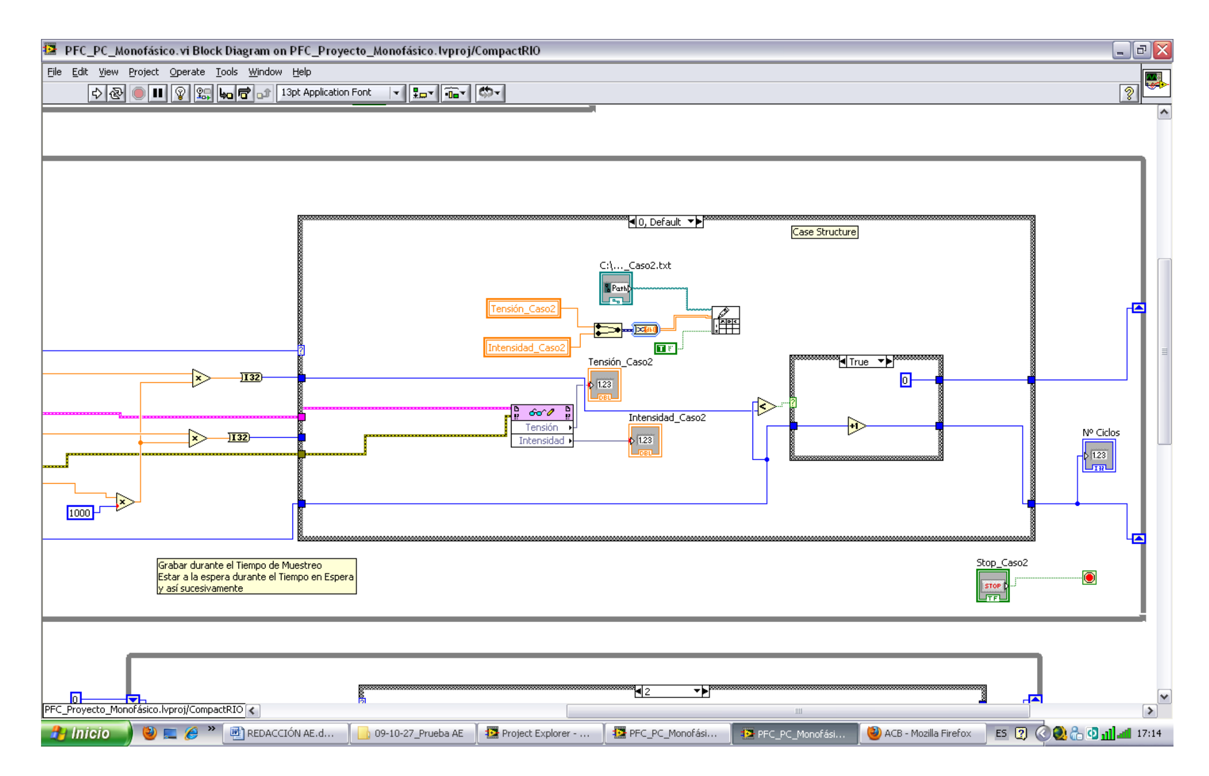


Figura 33: VI del segundo caso_caso 0_true_Monofásico.

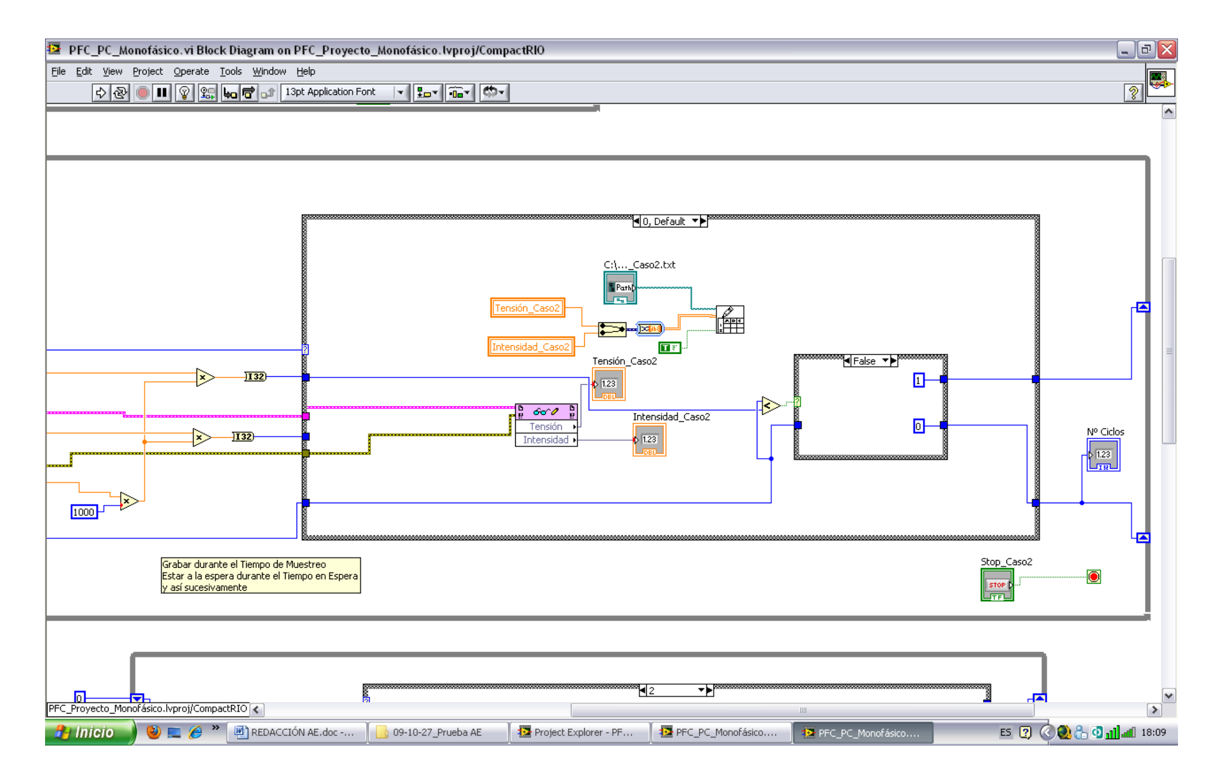


Figura 34: VI del segundo caso_caso 0_false_Monofásico.

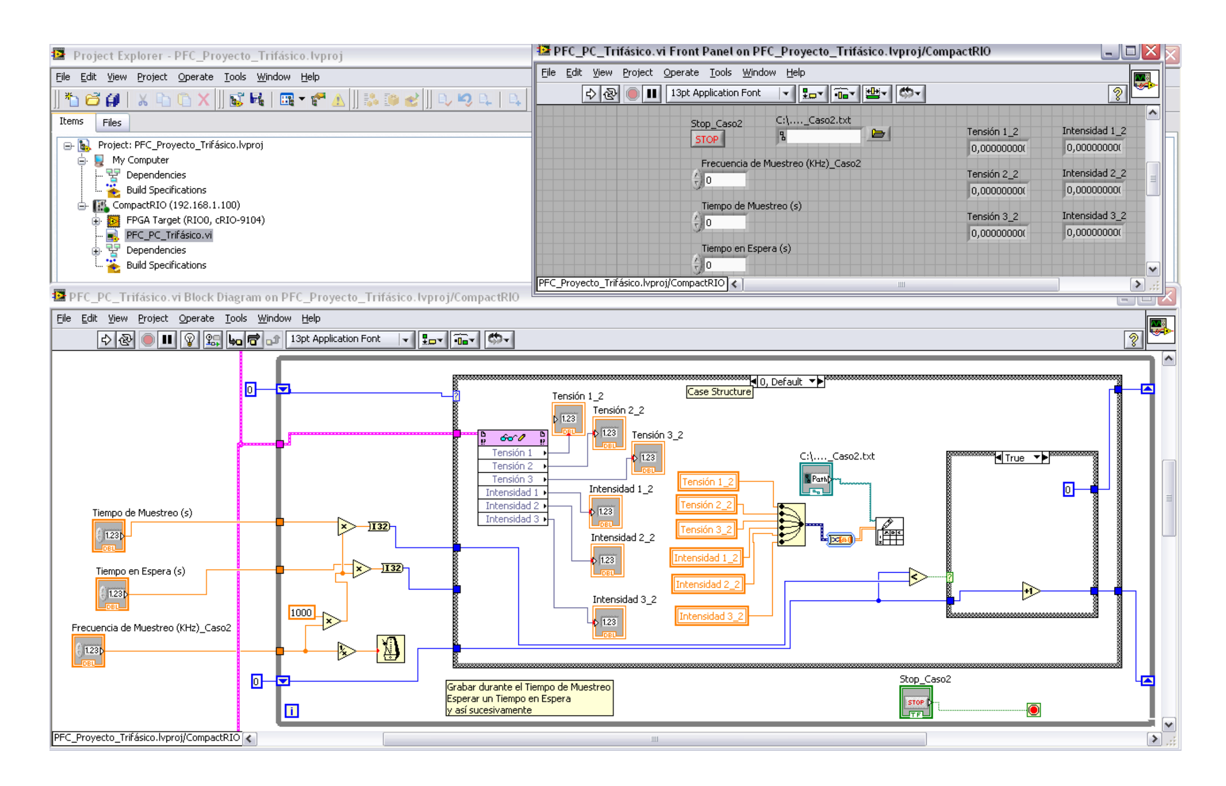


Figura 35: VI del segundo caso_caso 0_true_Trifásico.

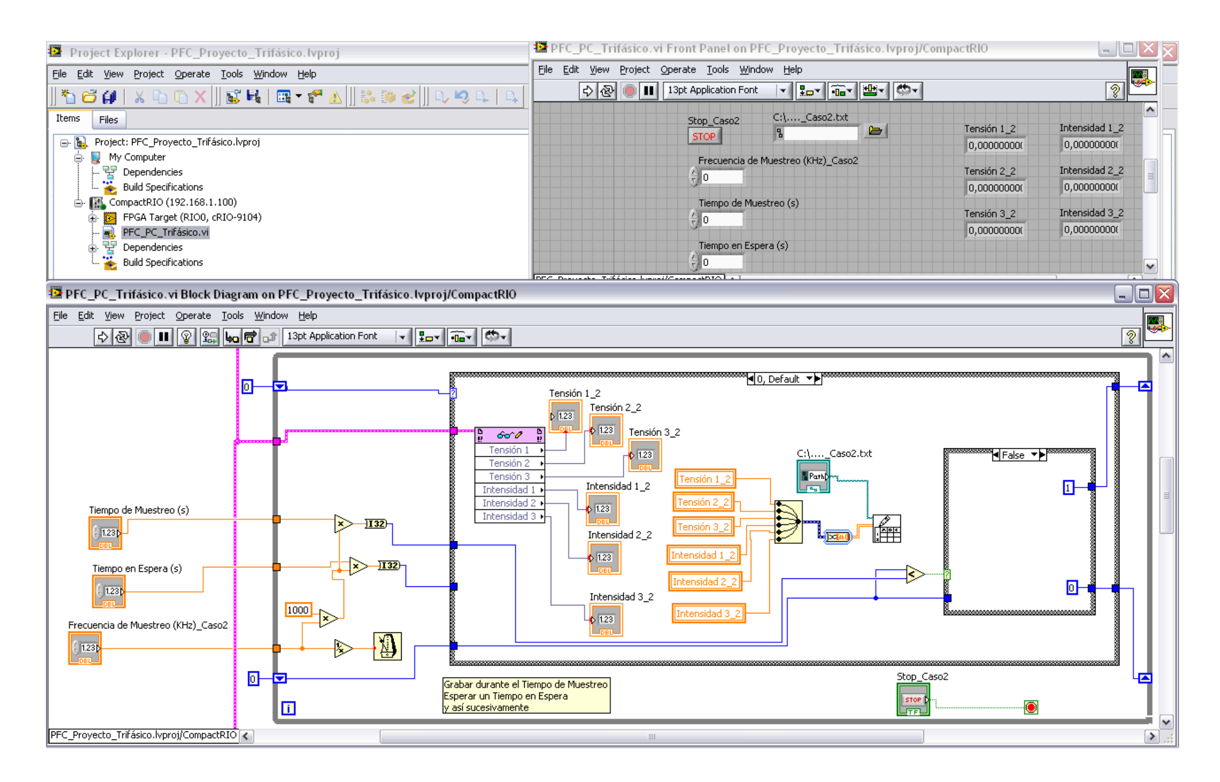


Figura 36: VI del segundo caso_caso 0_false_Trifásico.

El caso 1 corresponde a la situación de permanecer en espera. Como se pude ver en las figuras: Figura 32, Figura 37, Figura 38 Figura 39 y Figura 40 el usuario también introduce el tiempo en espera que junto con la

frecuencia de muestreo obtenemos el número de datos correspondiente a este caso. Como en el caso anterior, con una estructura "Case Structure" establecemos las comparaciones/relaciones que gobiernan la necesidad de seguir en este caso 1 o por el contrario pasar al caso 0 otra vez y comenzar de nuevo, ya que habrá transcurrido el tiempo en espera, y así sucesivamente hasta que el usuario pulse el "Control" de parada (Stop).

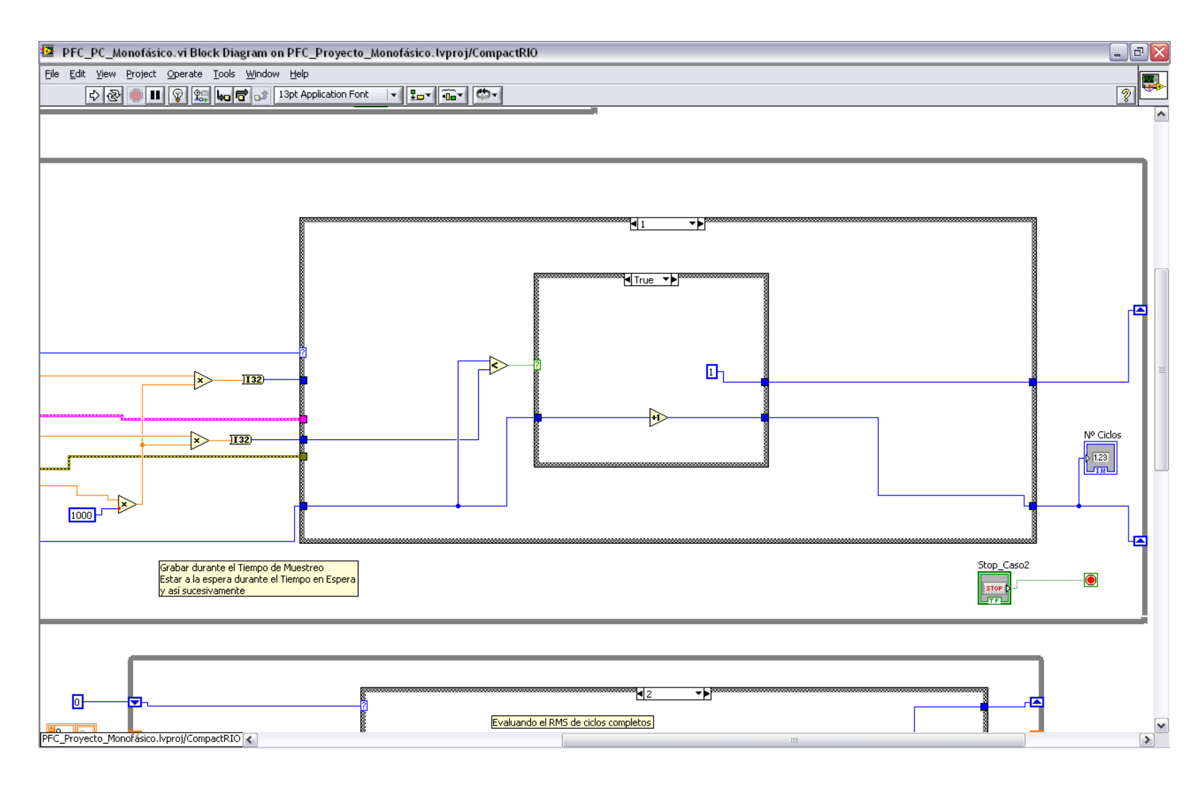


Figura 37: VI del segundo caso_caso 1_true_Monofásico.

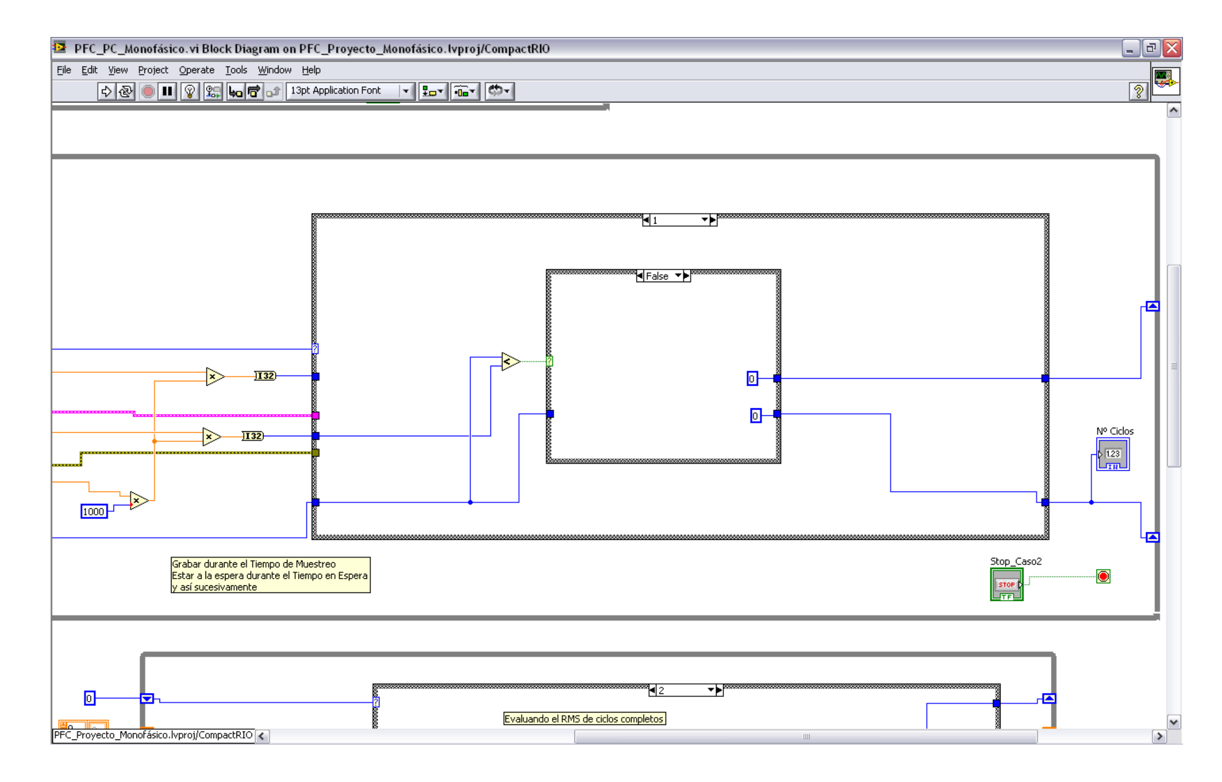


Figura 38: VI del segundo caso_caso 1_false_Monofásico.

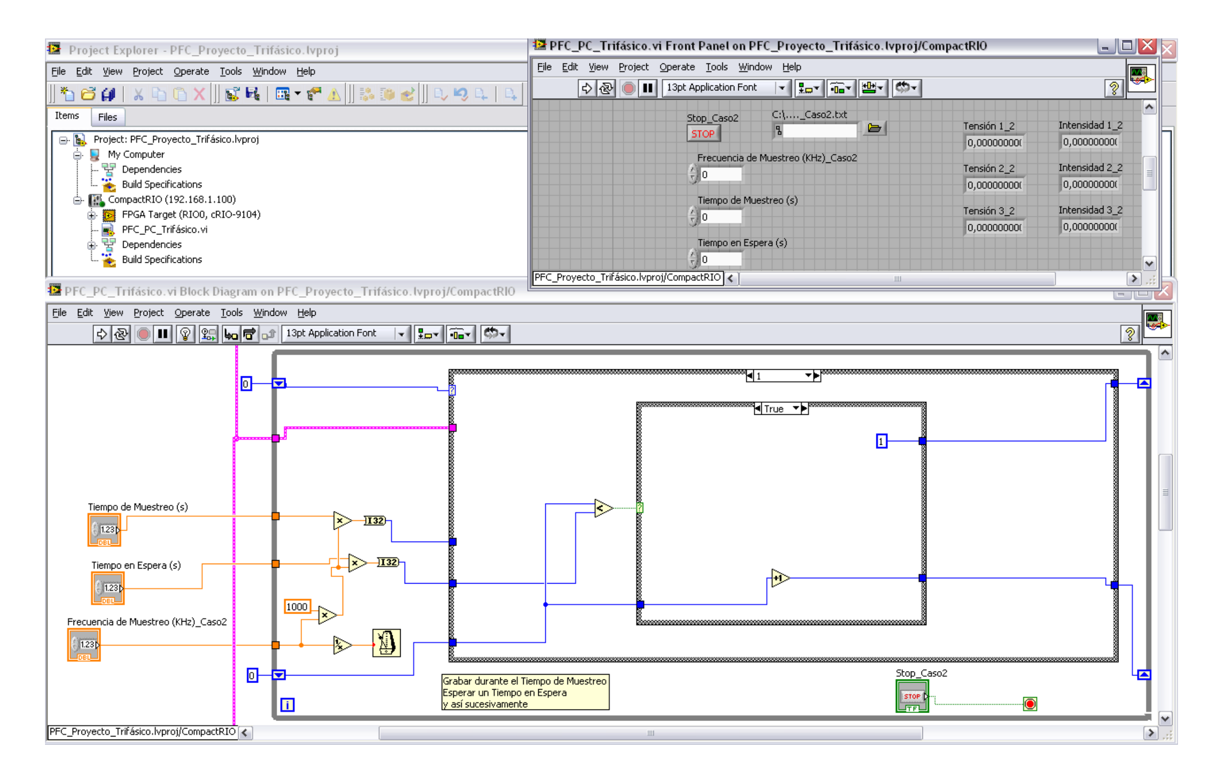


Figura 39: VI del segundo caso_caso 1_true_Trifásico.

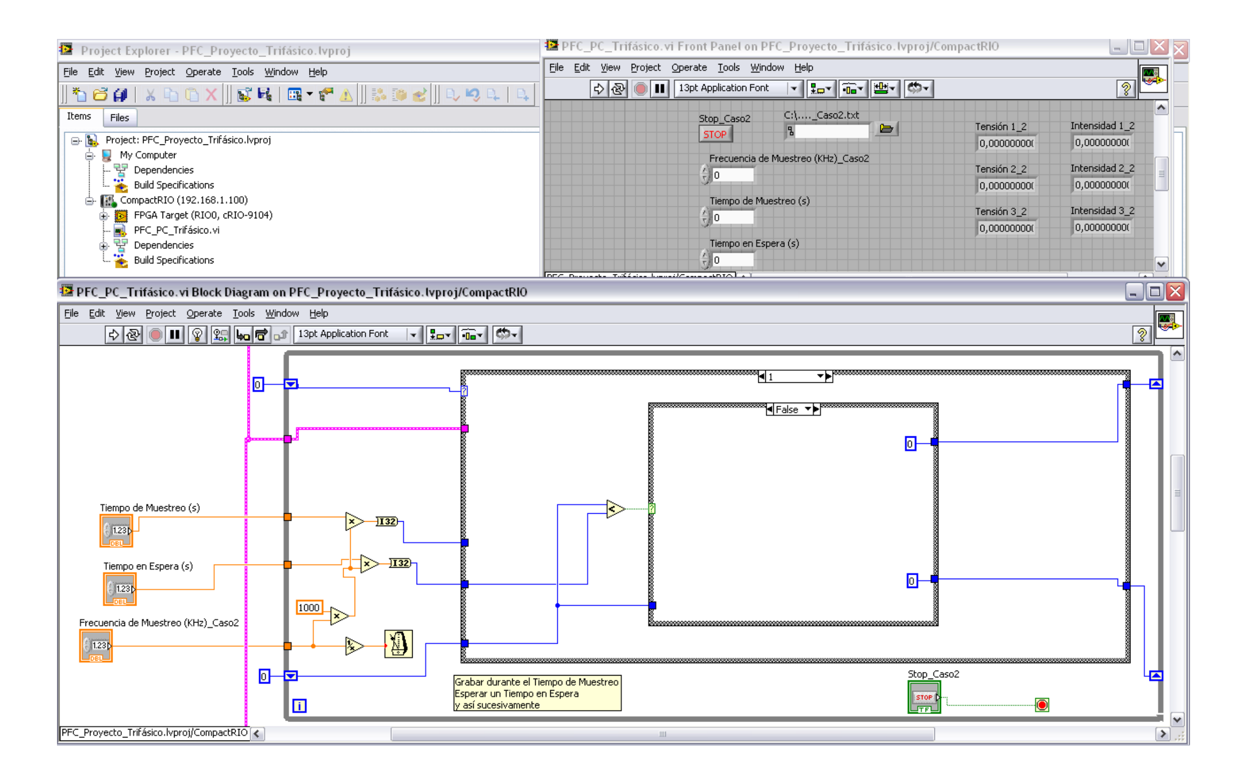


Figura 40: VI del segundo caso_caso 1_false_Trifásico.

Grabar un buffer de datos cuando se cumpla una condición especificada por el usuario

En este caso el usuario tiene que introducir un gran número de parámetros. Como en los casos anteriores se puede eligir la frecuencia de muestreo, el nombre/ubicación del archivo a guardar y accionar el botón de parada. Pero también hay que determinar la frecuencia de la red (que en España es de 50 Hz), el tiempo pre-evento, el tiempo post-evento, el umbral % de RMS<…, el RMS_Nominal, la tensión del primario y la tensión del secundario. Véase la Figura 41 y la Figura 42.

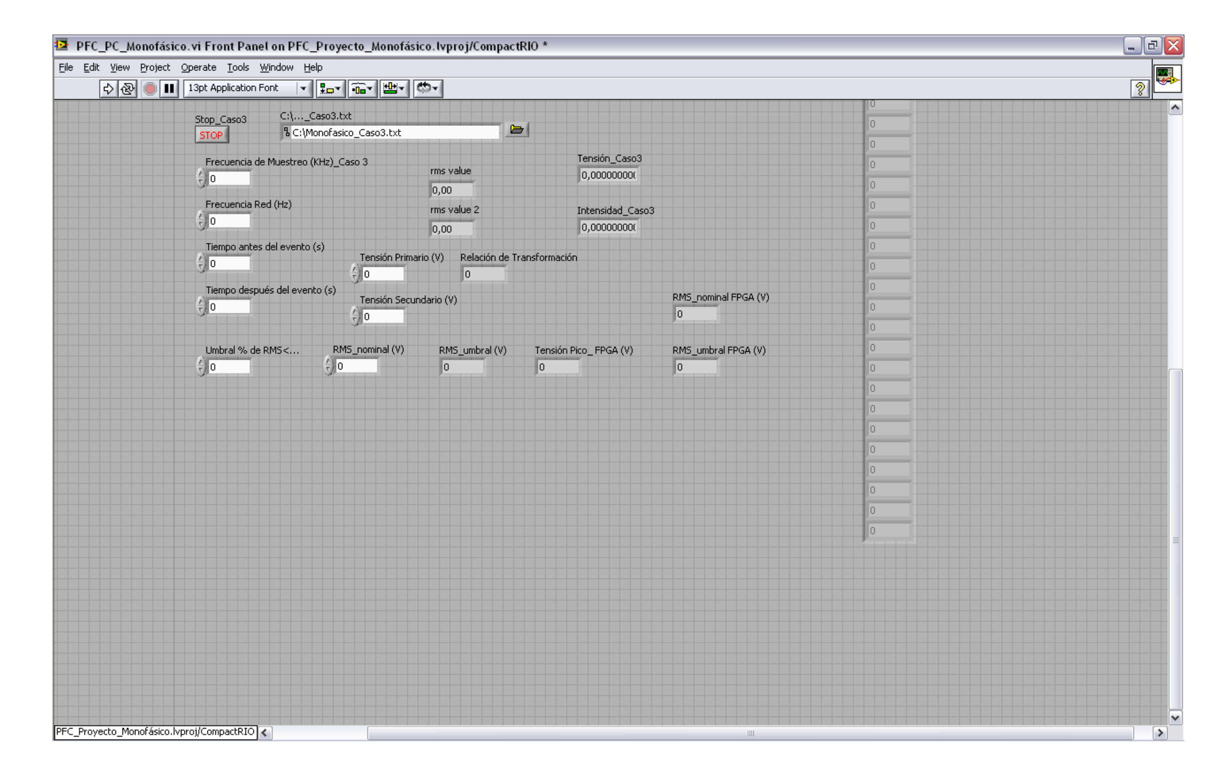


Figura 41: VI del tercer caso_vista general_Panel Frontal_Monofásico.

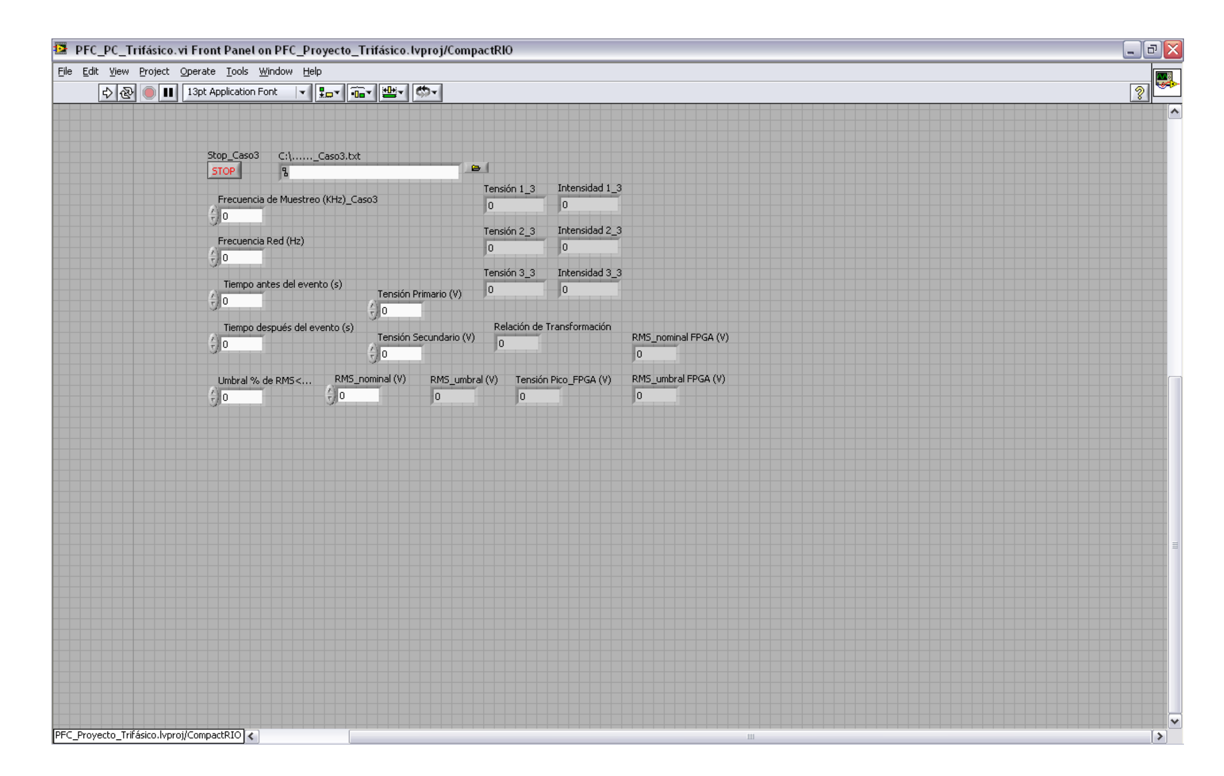


Figura 42: VI del tercer caso_vista general_Panel Frontal_Trifásico.

Como ya ha sido explicado en el tercer punto del apartado .Introducción y requerimientos la programación tiene que conseguir una serie de objetivos. Para conseguirlos necesitamos crear dos “array”(véase la

Figura 43 y la Figura 44), uno para ir tomando datos creando así un buffer de datos como si fuera una ventana deslizante, para obtener los datos correspondientes al periodo pre-evento y los ciclos evento. Una vez obtenidos se copian en el otro array y este array se inicializa para crear un buffer de datos correspondientes al periodo post-evento. Una vez obtenidos estos datos se añaden al array a grabar y el array usado como buffer se inicializa y el array a grabar también se inicializará una vez sean grabados los datos en archivo determinado por el usuario.

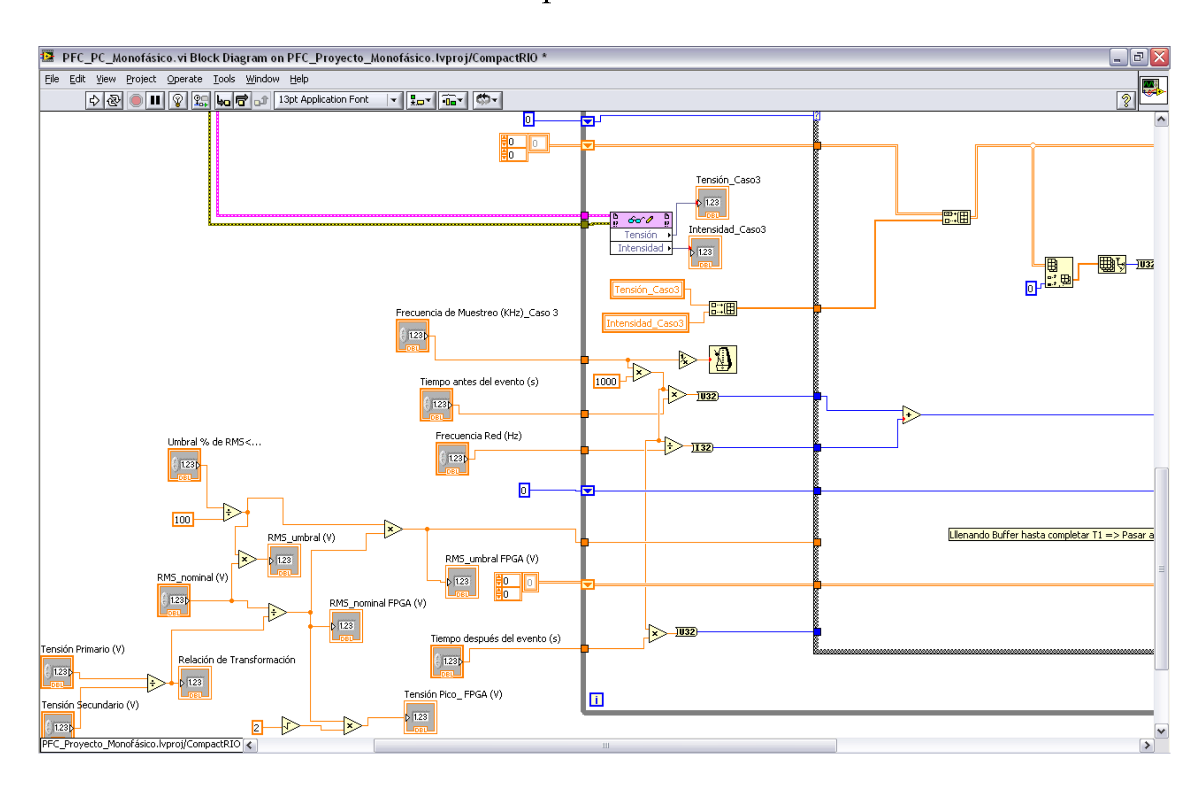


Figura 43: VI del tercer caso_vista general_Diagrama de Bloques_Monofásico.

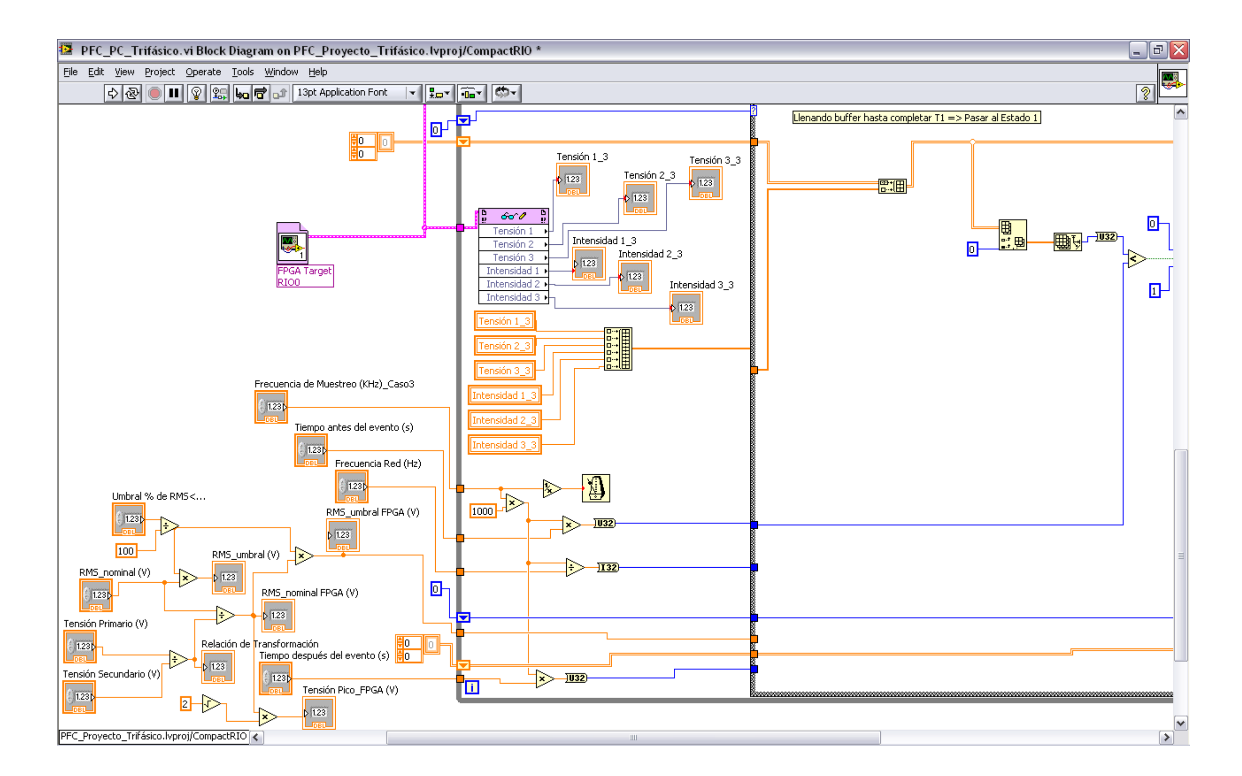


Figura 44: VI del tercer caso_vista general_Diagrama de Bloques_Trifásico.

Para conseguir todo hemos utilizado una estructura "Case Structure" con cinco casos del cero al cuarto.

En el caso 0 (Figura 45 y Figura 46) el programa está llenando de datos el array que se encuentra en la parte superior de la pantalla con la frecuencia de muestreo elegida, mientras el otro array permanece vacío. El programa permanecerá en este estado 0 hasta que no se coja el número de datos correspondiente al periodo pre-evento más un ciclo completo de onda, para este fin se realiza una comparación entre el número de datos que tiene el array y el número de datos correspondiente a este periodo más un ciclo y si es menor se permanece en el estado 0 y si no es así se pasará al estado 1.

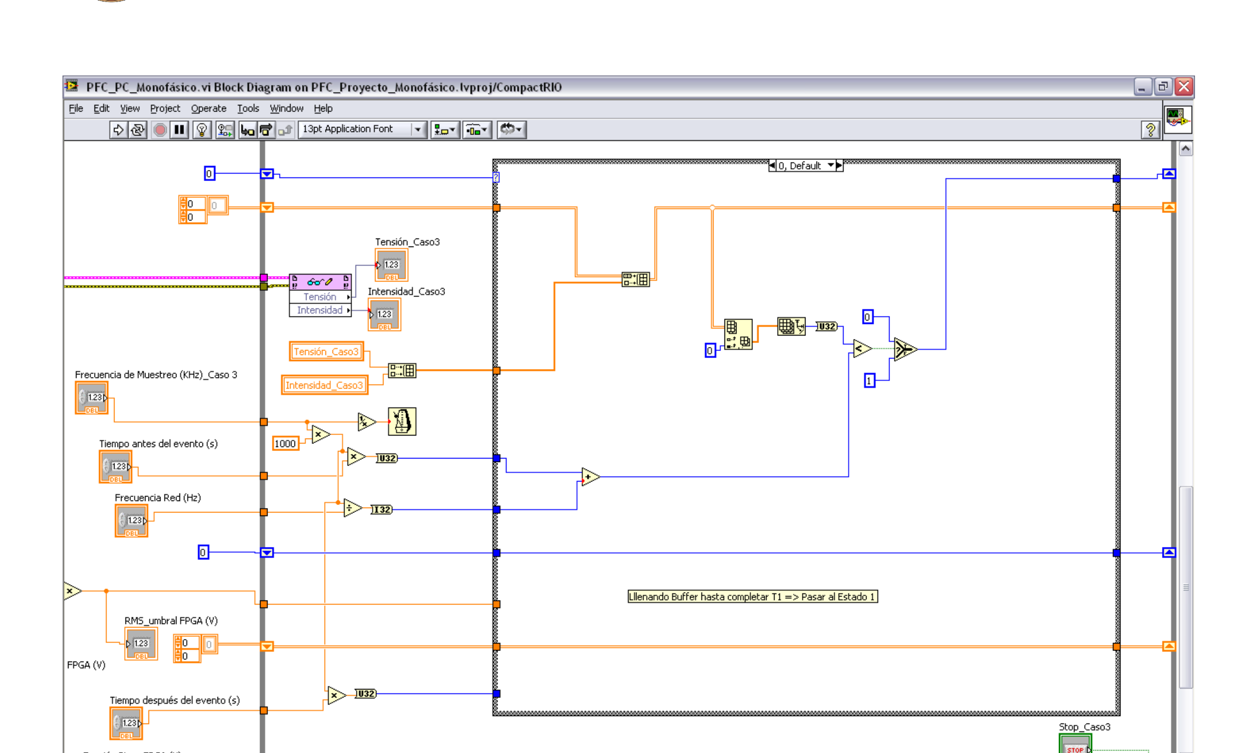


Figura 45: VI del tercer caso_caso 0_Monofásico.

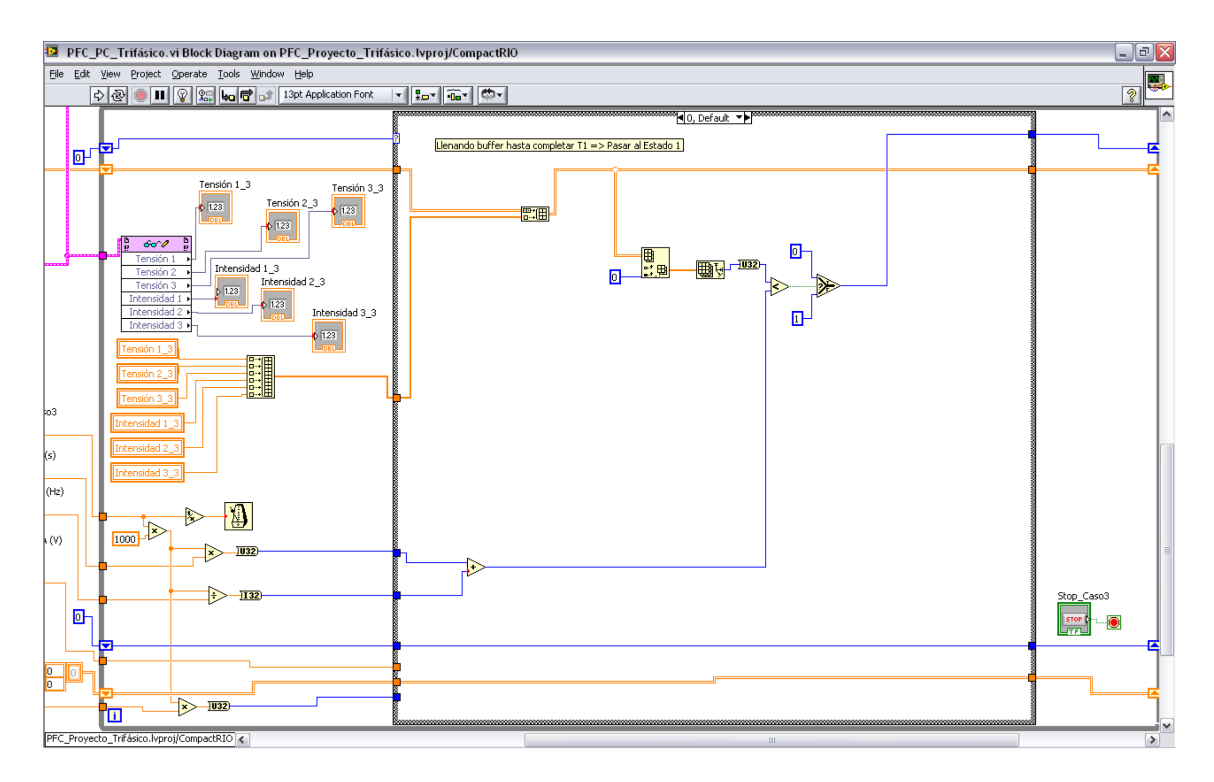


Figura 46: VI del tercer caso_caso 0_Trifásico.

En el estado 1 el programa está realizando de una forma continua lo siguiente. Cada ciclo o iteración de la estructura “While Loop” se coge un dato por cada canal, se entiende, y estos datos sustituyen al primer dato de

cada columna del array superior, y posteriormente se rotan los datos dentro del array y se coloca el nuevo dato en la última posición. Esto aunque parece rebuscado es lo más sencillo porque como el tamaño del array de datos dependerá de la frecuencia de muestreo, la frecuencia de la red y el periodo pre-evento determinados por el usuario a priori no sabemos su tamaño, por lo que lo mejor es introducir el nuevo dato en la posición 0 y posteriormente rotar los datos dentro del array, así también de un solo paso eliminamos el dato más antiguo. En la parte baja de la pantalla se puede ver como se cogen los datos correspondientes al último ciclo completo de onda y se evalúa su RMS si no es menor que la condición elegida se permanece en el caso 1, el array inferior seguirá vacío y se volverá a tomar un nuevo dato (véase la Figura 47 y la Figura 48) para repetir todo lo explicado, si por el contrario su RMS es menor que la condición (véase la Figura 49 y la Figura 50) todos los datos que hay en el array superior se copian en el array inferior y éste array superior se inicializa y se pasa al estado 2. Téngase en cuenta que en el sistema trifásico basta que una de las tres fases cumpla con la condición para considerarlo un ciclo evento y proseguimos evaluando la fase que falla como las otras dos.

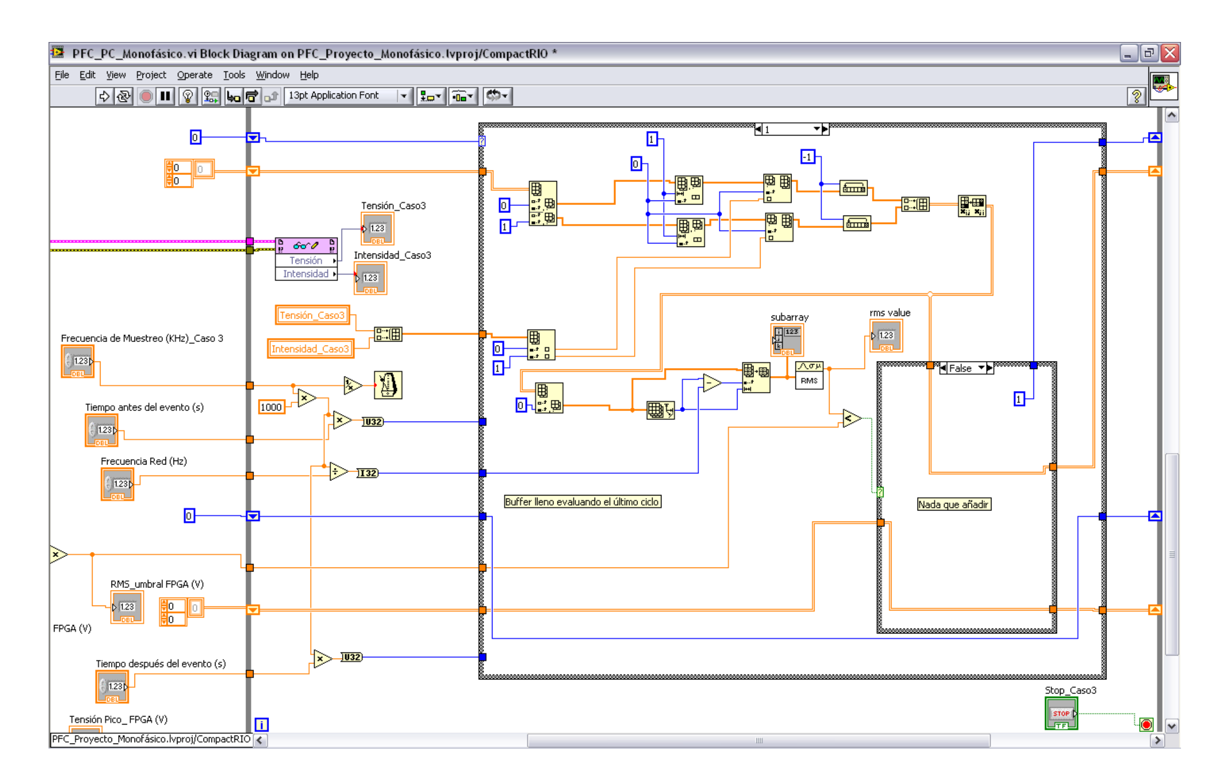


Figura 47: VI del tercer caso_caso 1_false_Monofásico.

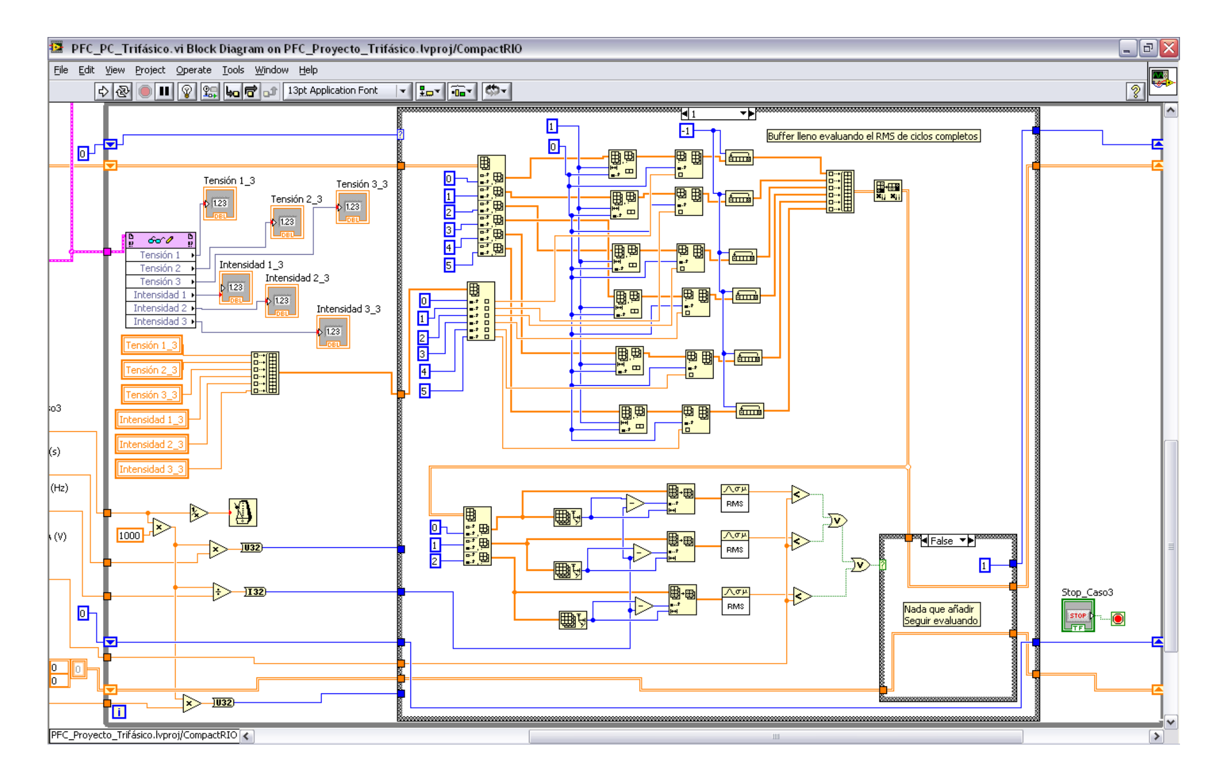


Figura 48: VI del tercer caso_caso 1_false_Trifásico.

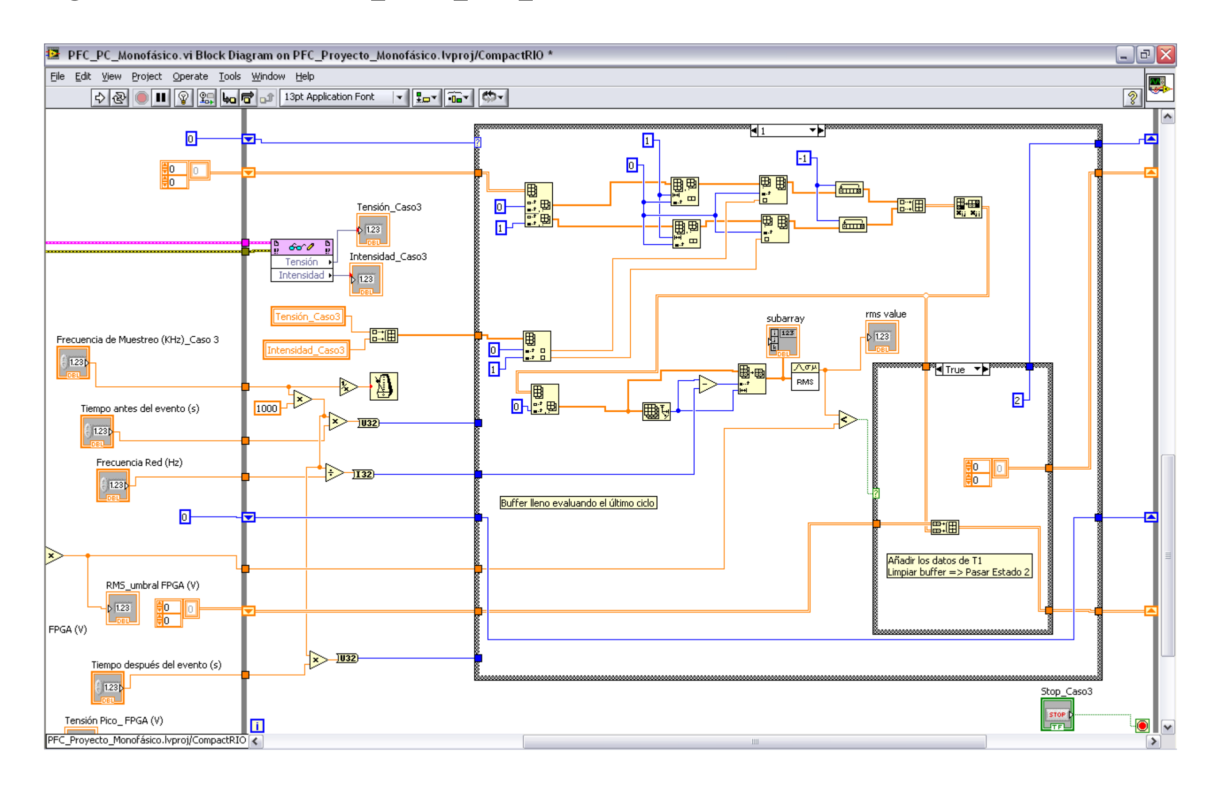


Figura 49: VI del tercer caso_caso 1_true_Monofásico.

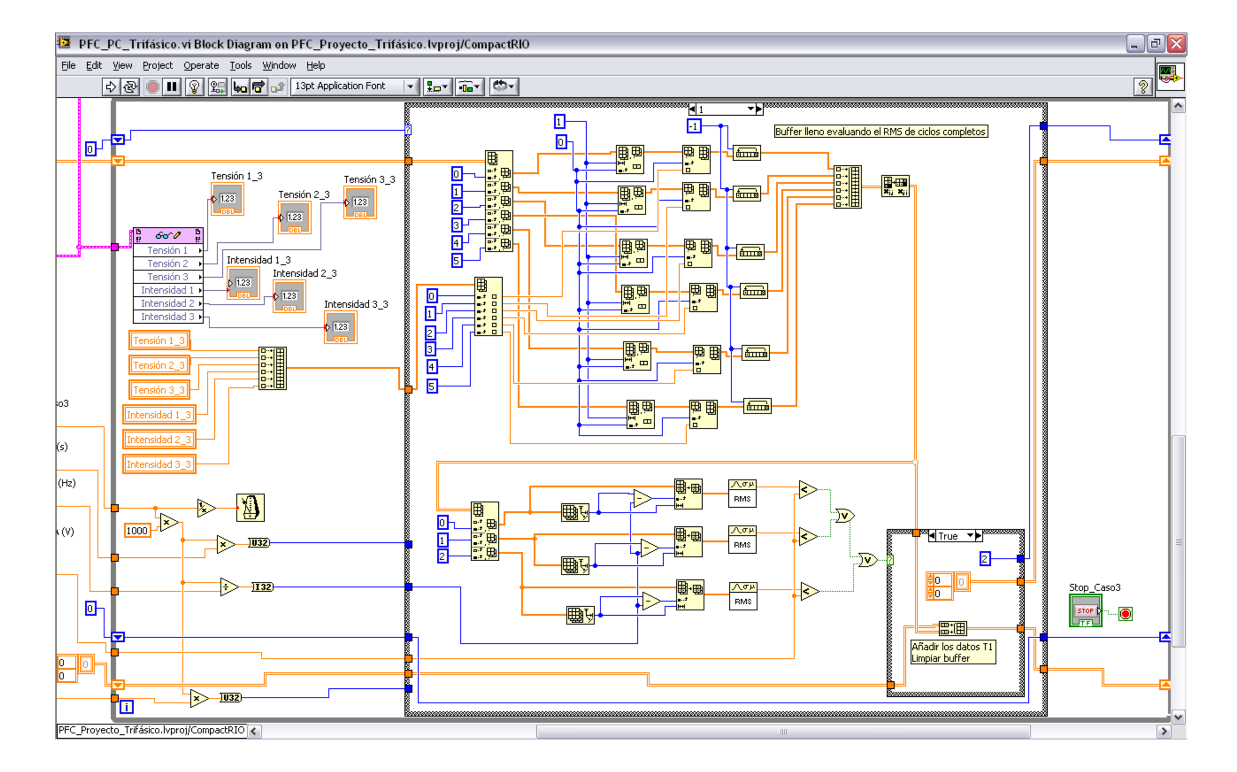


Figura 50: VI del tercer caso_caso 1_true_Trifásico.

En el caso 2 se toman datos y se incorporan al array superior que recordemos que está vacío en este momento (Figura 51 y Figura 52), hasta que el número de datos en el array corresponde a un ciclo completo de onda. Una vez que tengamos los datos correspondientes a un ciclo pasamos al “false case” de la estructura “Case Structure” (Figura 53, Figura 54, Figura 55 y Figura 56).

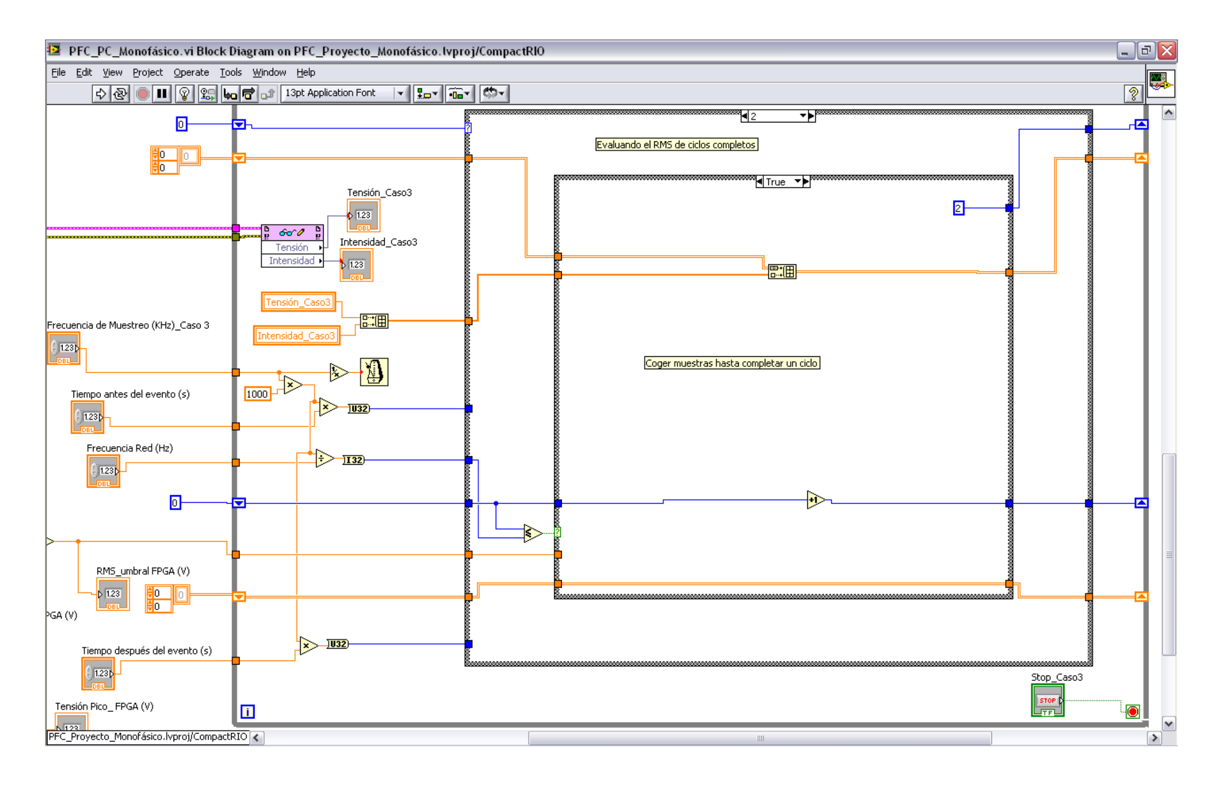


Figura 51: VI del tercer caso_caso 2_true_Monofásico.

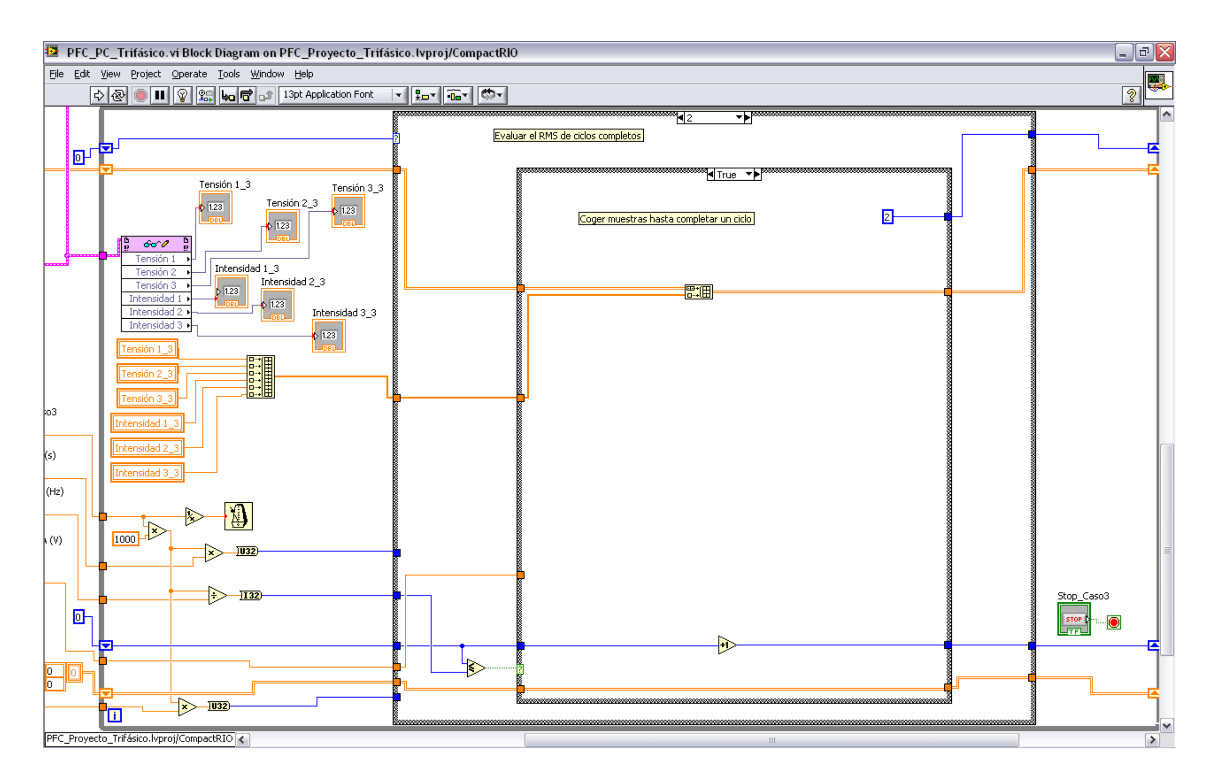


Figura 52: VI del tercer caso_caso 2_true_Trifásico.

Ahora se evalúa el RMS de este ciclo, si su valor es menor que la condición exigida (véase la Figura 53 y la Figura 54) los datos se incorporan al array inferior que recordamos que contiene los datos

correspondientes al periodo pre- evento, al primer ciclo evento y ahora también al segundo ciclo evento; y se vuelve a inicializar el array superior y el programa se mantendrá en el caso 2. En el sistema trifásico se sigue el proceso con las tres fases. Como se ha inicializado el array y se mantiene en el caso 2, como no tenemos datos en el array estaremos otra vez en el caso anterior (Caso 2_true, es decir Figura 51 y Figura 52). Si por el contrario el valor RMS del ciclo que estamos evaluando es mayor que la condición pasamos al estado 3, se inicializa el array superior y no se incorpora los datos de este ciclo al array inferior (Figura 55 y Figura 56). Esta situación se puede presentar una vez que tengamos un ciclo evento o cuando tengamos enésimos ciclos eventos el funcionamiento es el mismo.

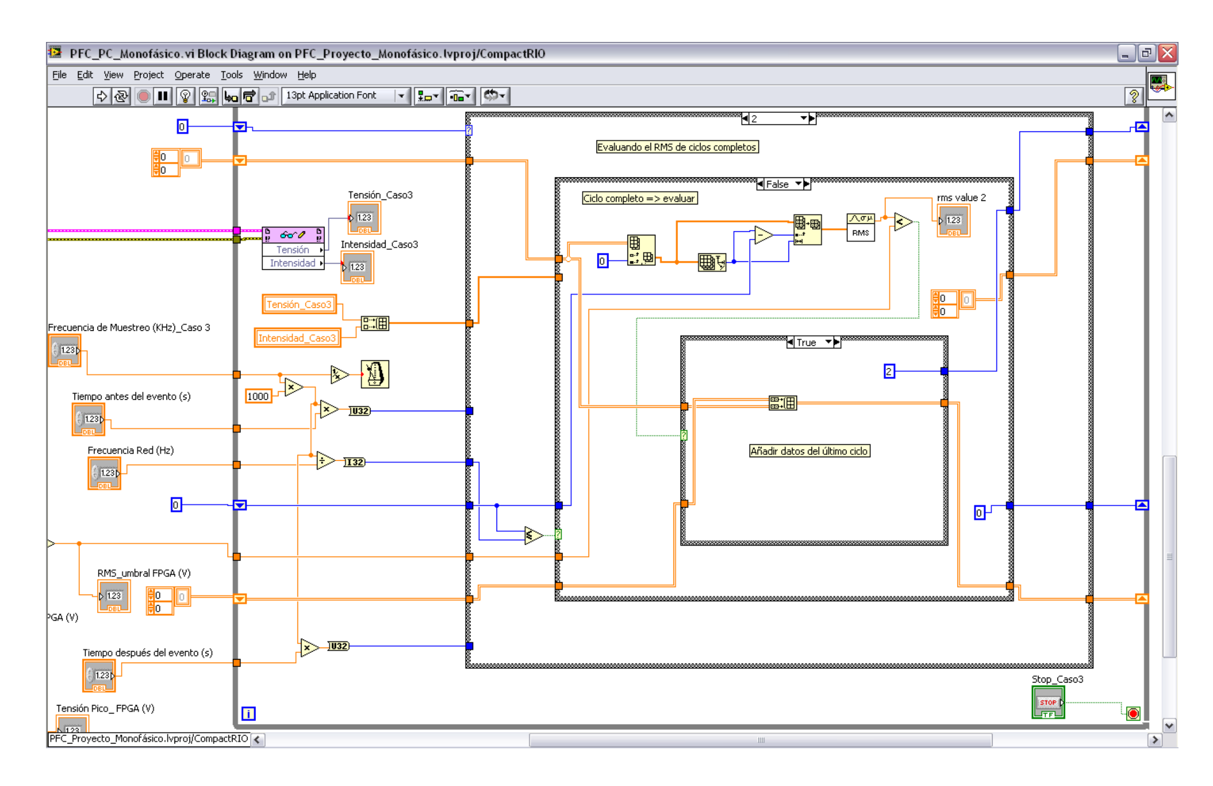


Figura 53: VI del tercer caso_caso 2_false_true_Monofásico.

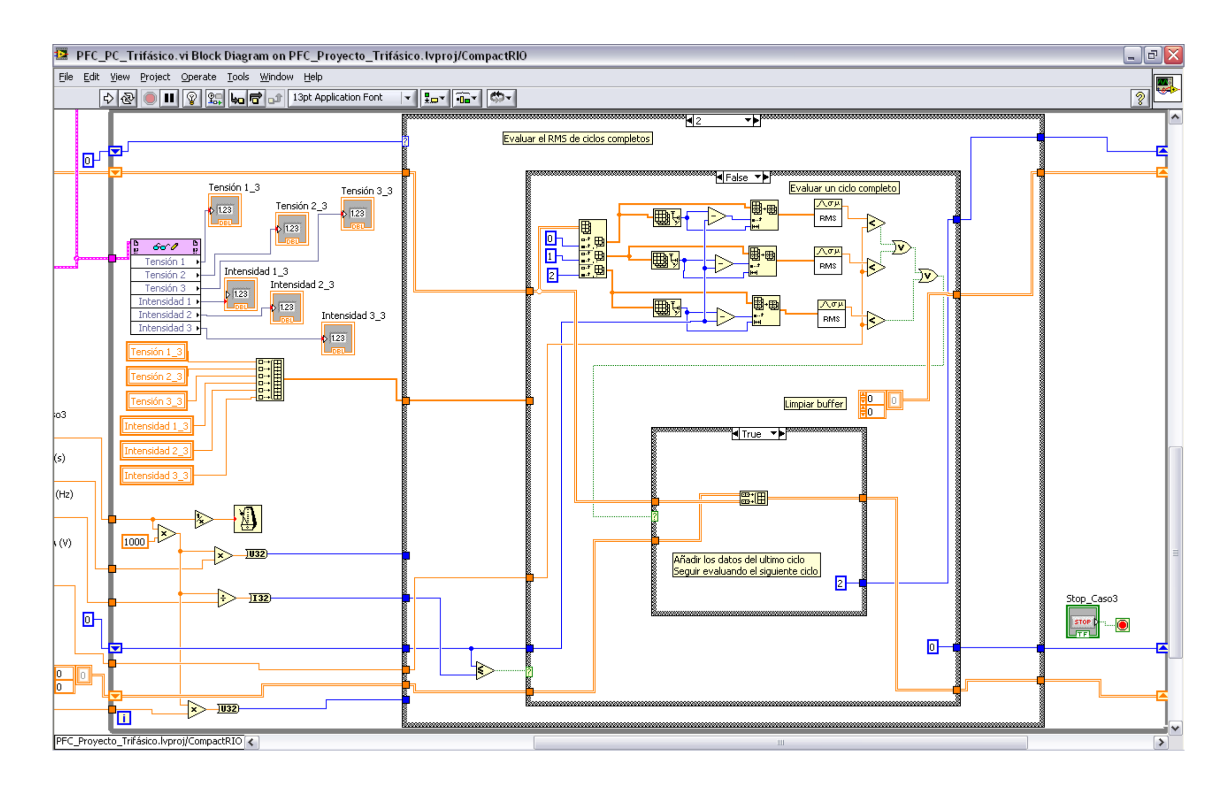


Figura 54: VI del tercer caso_caso 2_false_true_Trifásico.

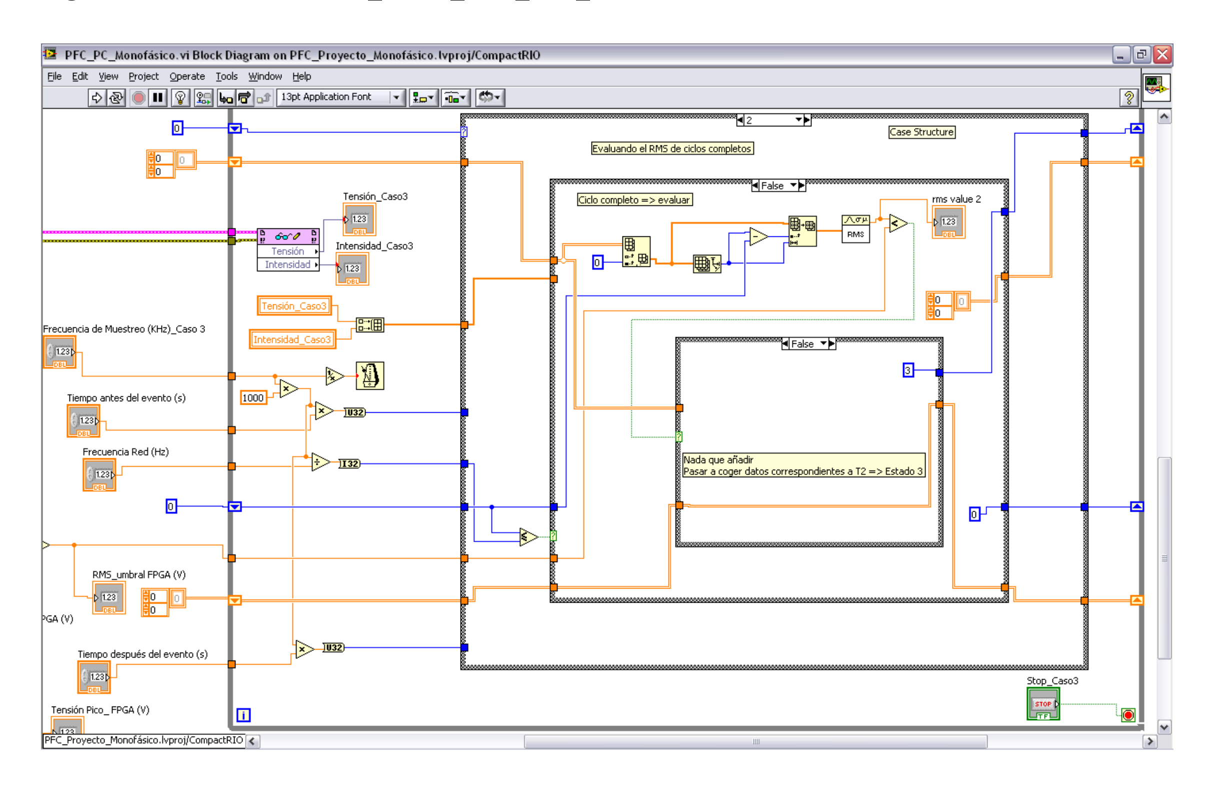


Figura 55: VI del tercer caso_caso 2_false_false_Monofásico.

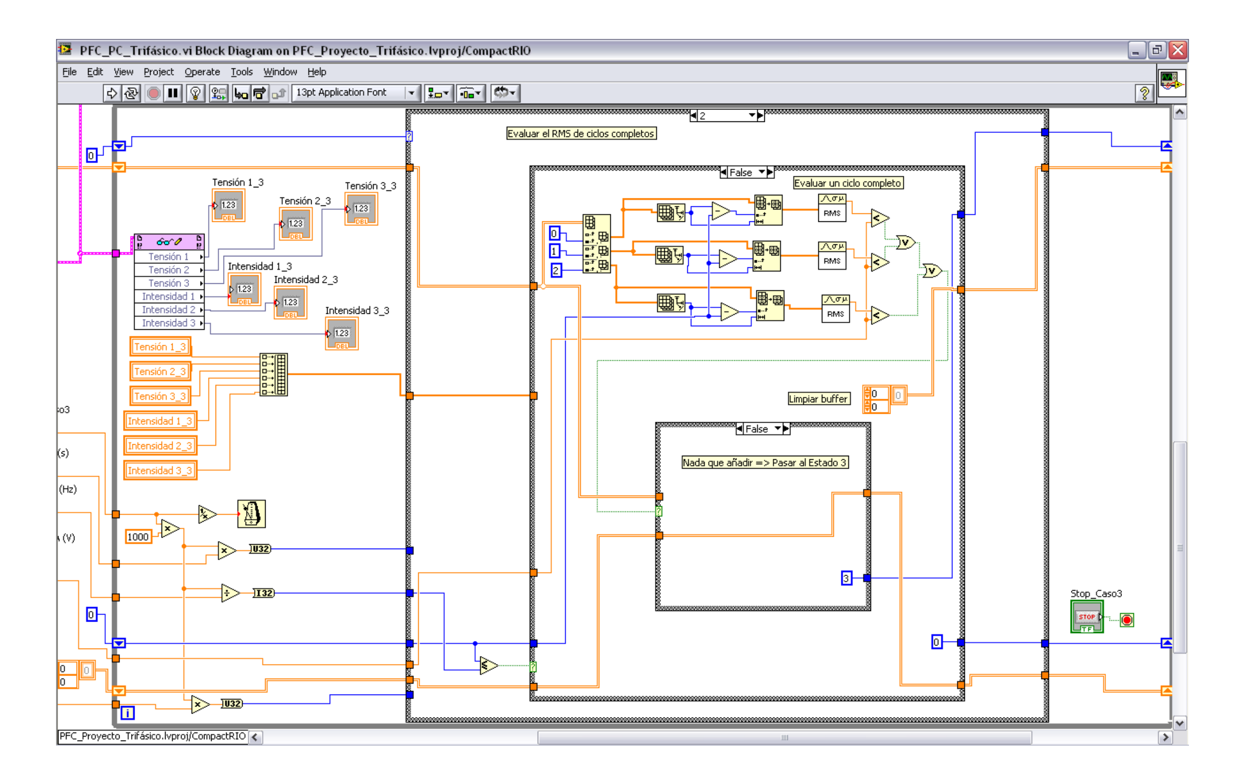


Figura 56: VI del tercer caso_caso 2_false_false_Trifásico.

En el estado 3 simplemente incorporamos datos al array superior (Figura 57 y Figura 58) (recodemos que está vacío en este momento) hasta que tengamos el número de datos correspondientes al periodo post-evento. Una vez que tengamos los datos (véase la Figura 59 y la Figura 60) éstos se incorporan al array inferior que por lo tanto ya tendrá todos los datos a guardar que son los correspondientes al periodo pre-evento, ciclo/s evento/s y al periodo post-evento. Se inicializa el array superior y pasamos al caso 4.

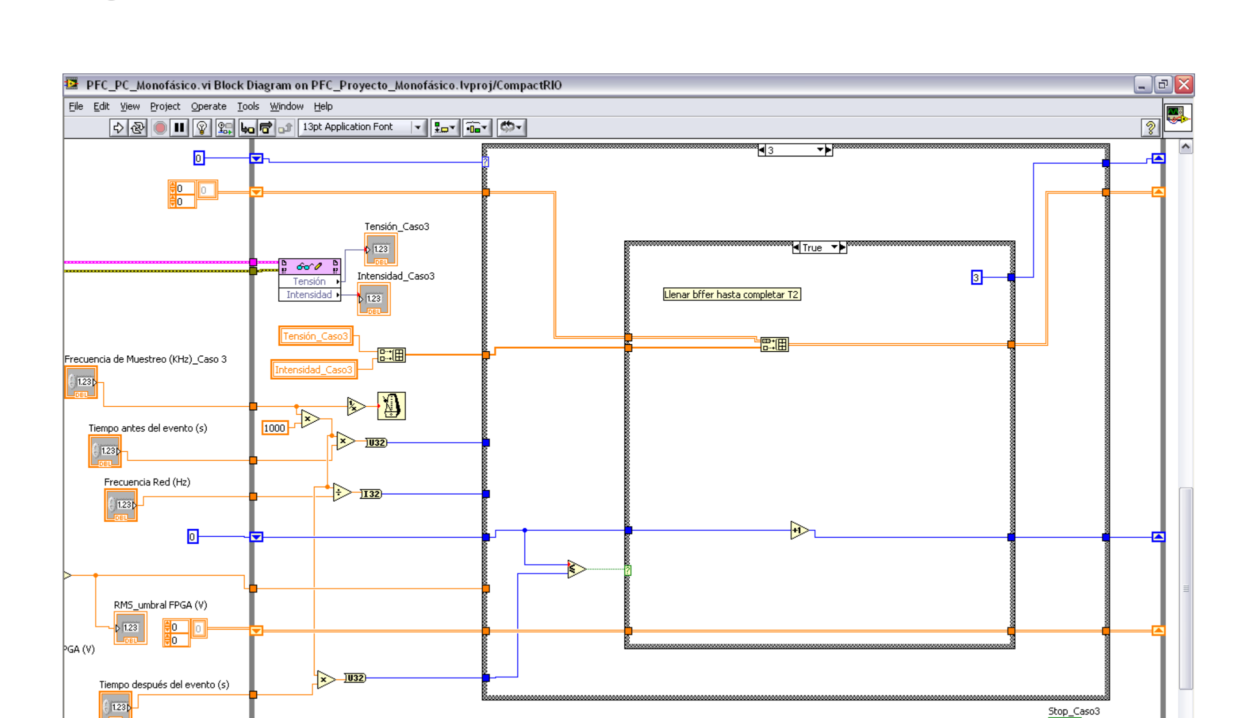


Figura 57: VI del tercer caso_caso 3_true_Monofásico.

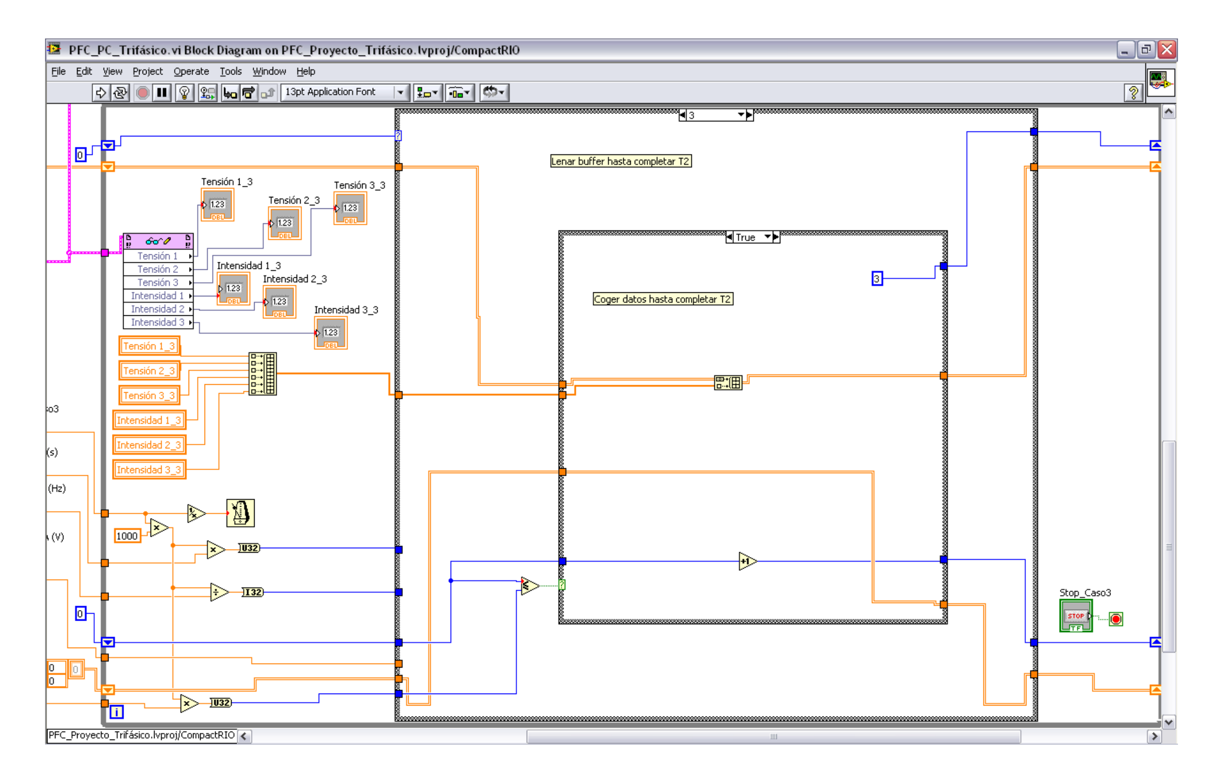


Figura 58: VI del tercer caso_caso 3_true_Trifásico.

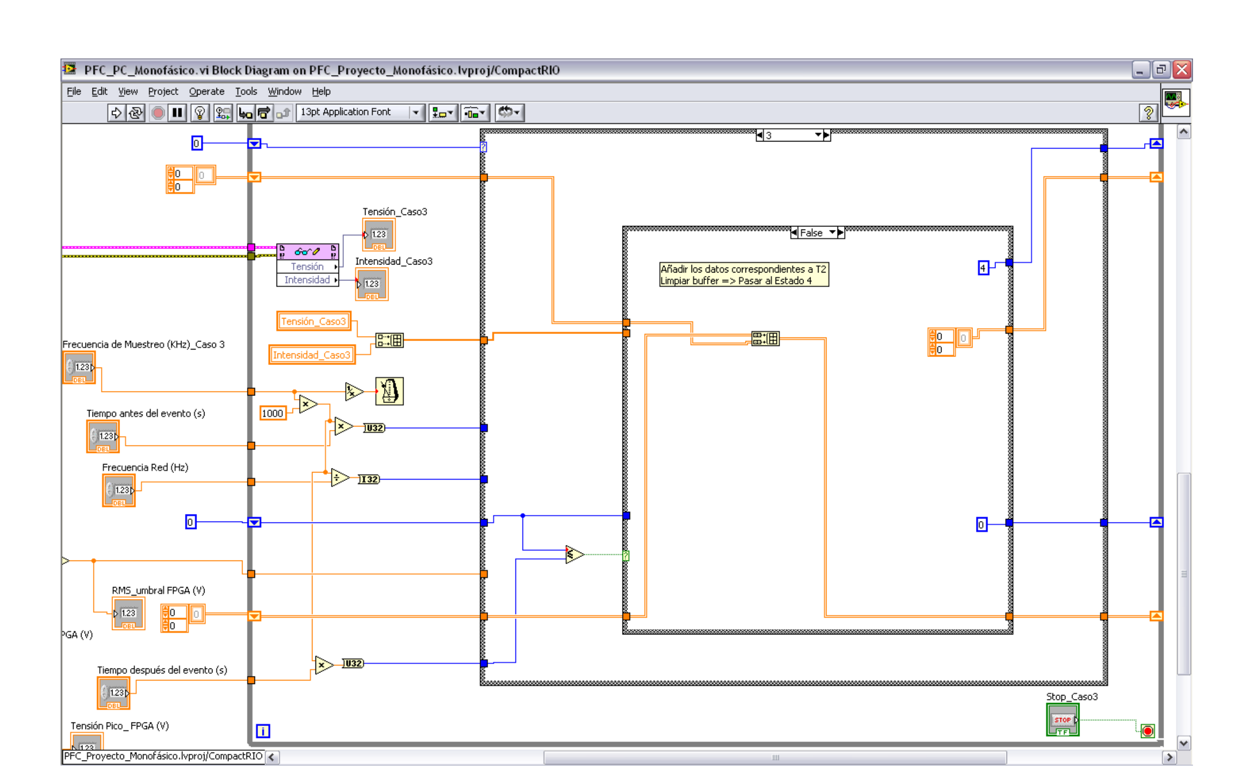


Figura 59: VI del tercer caso_caso 3_false_Monofásico.

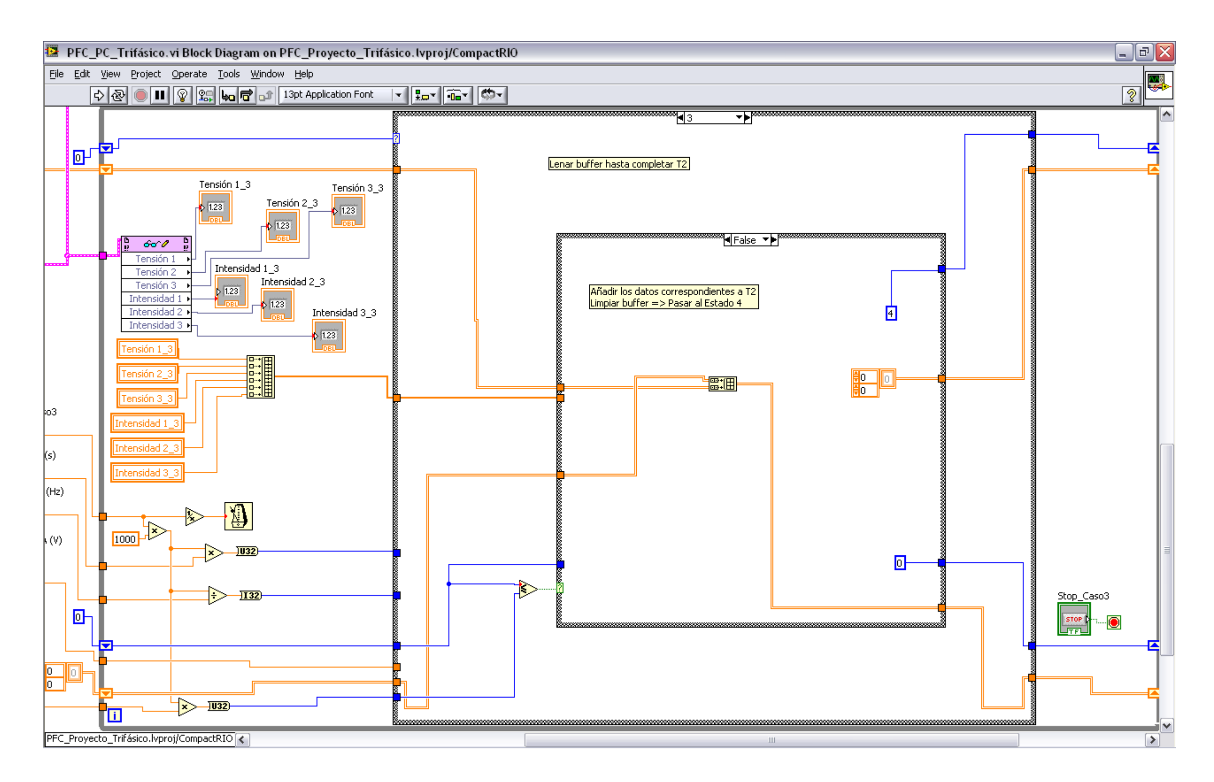


Figura 60: VI del tercer caso_caso 3_false_Trifásico.

El caso 4 (Figura 61 y Figura 62) sirve para grabar los datos que contiene el array inferior en la ubicación y con el nombre que determine el

usuario en el "Panel Frontal", y pasamos al estado cero otra vez y así sucesivamente hasta que el usuario pulse el botón de parada.

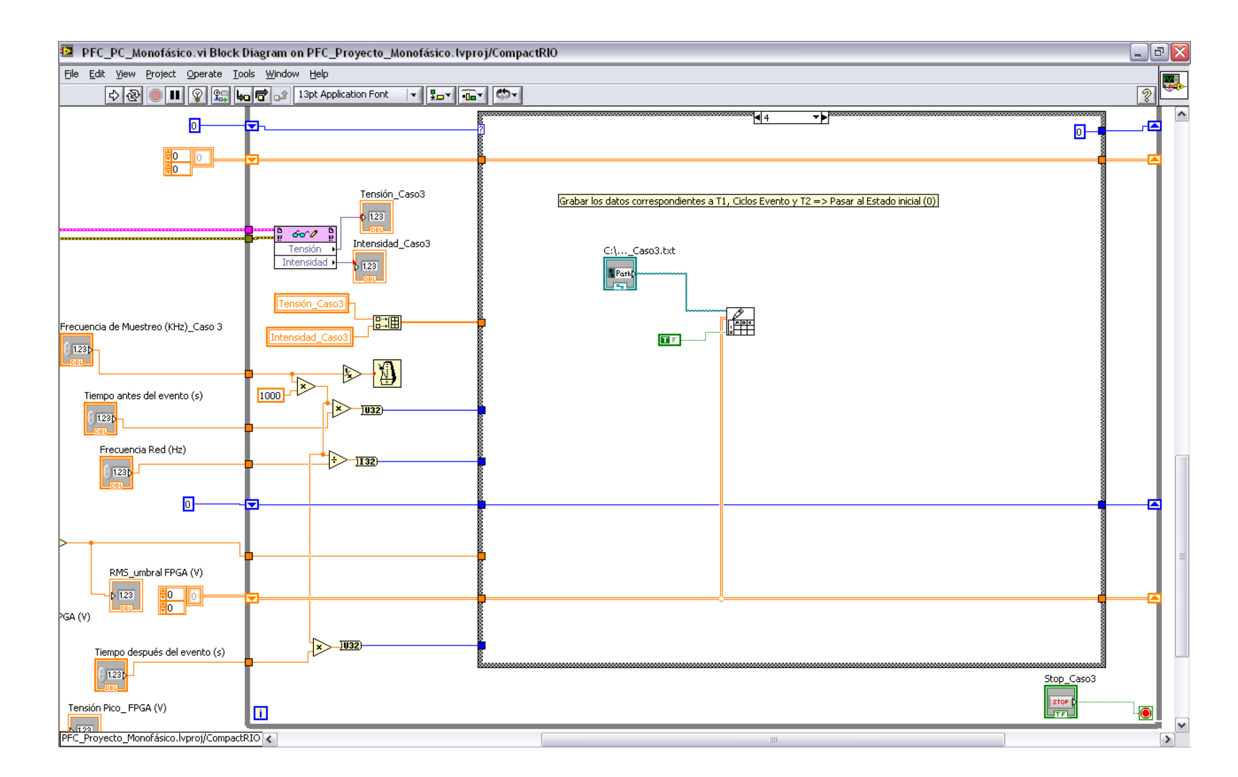


Figura 61: VI del tercer caso_caso 4_Monofásico.

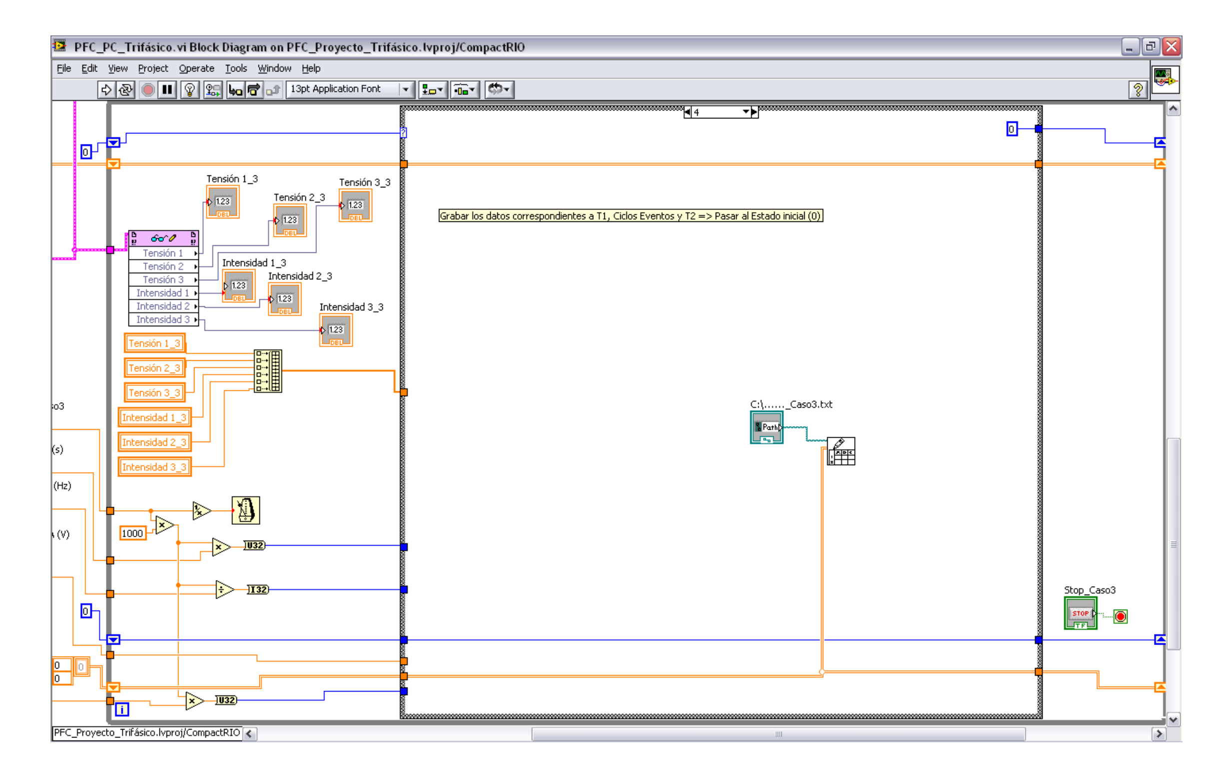


Figura 62: VI del tercer caso_caso 4_Trifásico.

Resultados y ensayos

Introducción: limitaciones

CompactRIO es un sistema avanzado de control y adquisición embebido motorizado por tecnología National Instruments (NI) reconfigurable (RIO). Los fabricantes de sistemas de adquisición y control usan ampliamente dispositivos de arreglos de compuertas programables de punto flotante (FPGA) debido a su rendimiento, posibilidad de reconfiguración, tamaño y bajos costos de desarrollo de ingeniería. El sistema de adquisición de datos CompactRIO es pequeño y extremadamente robusto.

Usando NI LabVIEW, los programadores no tienen que saber lenguajes especializados de diseño de hardware, tales como el VHDL, de los FPGAs para crear sistemas reconfigurables de control y adquisición de datos que sean altamente optimizados. VHDL es un lenguaje de programación que permite la programación de hardware de bajo nivel,tal como una compuerta lógica. Debido a su nivel de dificultad y complejidad, la programación de un FPGA toma mucho tiempo de desarrollo del proceso y se ha convertido en un asunto importante en las etapas de implementación.

Con CompactRIO uno puede construir rápidamente sistemas embebidos de control o de adquisición de datos que rivalizan con el rendimiento y optimización del hardware especializado.

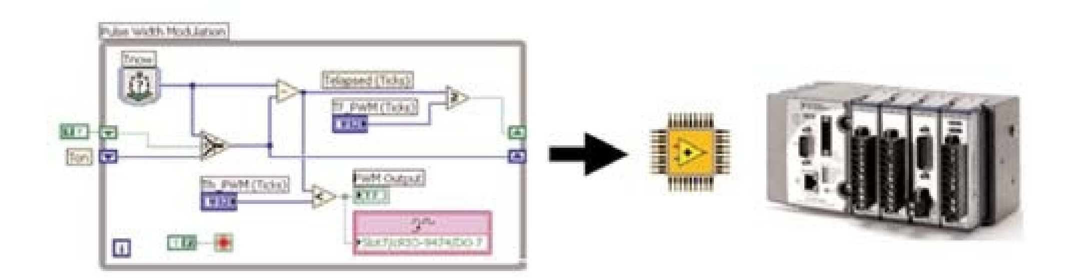
CompactRIO combina un bajo consumo y un procesador embebido de tiempo real con el gran rendimiento de un chip RIO FPGA. El núcleo RIO posee mecanismos de transferencia incorporados para transferir datos al procesador embebido para realizar análisis de tiempo real, post-procesamiento, data logging o comunicación con una computadora servidor en red. Provee acceso directo del hardware a los circuitos de E/S de cada módulo de E/S usando funciones elementales LabVIEW FPGA. Cada módulo incluye conectividad propia, acondicionamiento de señales, circuito de conversión (tal como ADC o DAC) y una barrera aislante opcional. Esto representa una arquitectura de bajo costo con acceso abierto a los recursos de bajo nivel del hardware.

Cada módulo de E/S del CompactRIO posee un acondicionamiento de señales y bornera de tornillos integrados, con conectores BNC o DSUB. Al integrar la bornera en los módulos, el sistema CompactRIO reduce significativamente los requerimientos de espacio y los costos de cableado. Existe una variedad de diversos tipos de E/S, que incluyen entradas de

termocupla de ± 80 mV, E/S analógicas de muestreo simultáneo de ±10 V (que son las utilizadas por nosotros), E/S digitales industriales de 24 V con intensidad de corriente de hasta 1 A, entradas digitales diferenciales/TTL con salida de fuente regulada de tensión de 5 V para codificadores y entradas digitales universales de 250 Vrms. Debido a que los módulos poseen acondicionamiento de señales integrado para rangos de voltaje extendidos o tipos de señales industriales, usualmente se pueden cablear las conexiones directamente desde el módulo CompactRIO a los sensores y actuadores.

Los sistemas embebidos CompactRIO poseen un procesador industrial del tipo Pentium de 200 MHz que ejecuta de manera confiable y determinística aplicaciones LabVIEW Real-Time. Se pueden usar miles de funciones de LabVIEW para construir un sistema embebido multi-hilo para control en tiempo real, análisis,data logging y comunicaciones.

El circuito FPGA es una máquina reconfigurable de procesamiento paralelo que ejecuta la aplicación hecha en LabVIEW sobre el circuito de silicio de un chip. Usando el hardware embebido RIO FPGA, se pueden implementar sistemas de control PID de múltiples hilos a velocidades superiores a los 100.000 muestras/s. Se pueden implementar sistemas de control digital a velocidades de lazos de hasta 1 MM/s y evaluar múltiples anillos de lógica Booleana usando lazos while de ciclo simple de hasta 40 MHz (25 ns). Debido a la naturaleza paralela del núcleo RIO, la adición de cálculo no necesariamente reduce la velocidad de la aplicación FPGA. CompactRIO ofrece chasis de cuatro y ocho ranuras con opciones de chips FPGA de 1 millón o 3 millones de compuertas.



Ensayo de toma manual de datos

Grabar los datos medidos o permanecer a la espera según lo estime oportuno el usuario.

En el siguiente ensayo la frecuencia de muestreo es de 40 KHz.

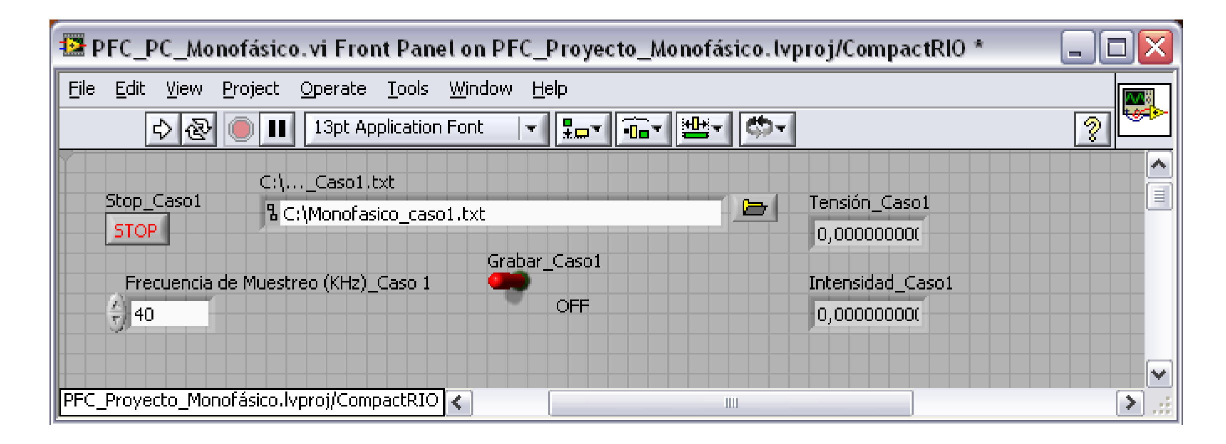


Figura 63: Panel Frontal_grabar OFF_Ensayo caso 1_Ensayo de toma manual de datos.

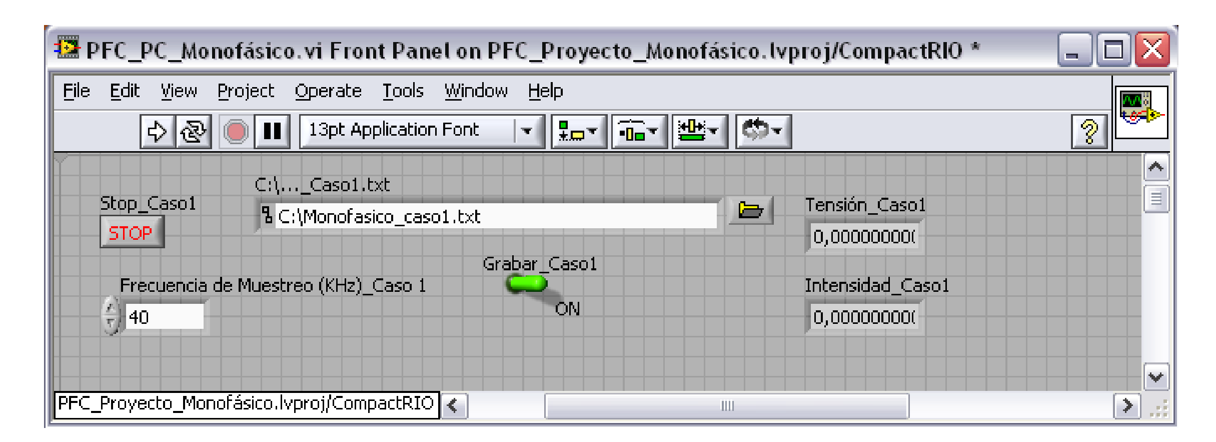


Figura 64: Panel Frontal_grabar ON_Ensayo caso 1_Ensayo de toma manual de datos.

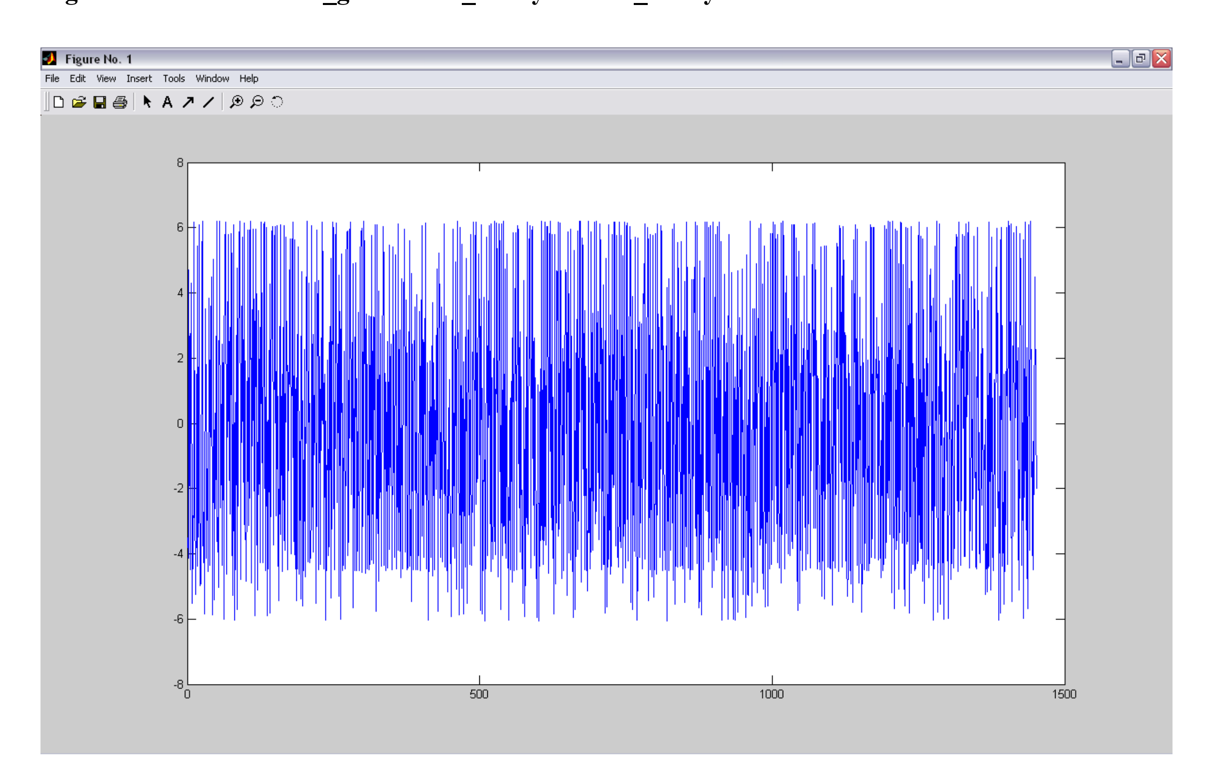


Figura 65: Ensayo caso 1_ Ensayo de toma manual de datos.

Simplemente en este ensayo grabamos datos unos segundos.

Ensayo de toma automática de datos

Grabar los datos durante un tiempo y permanecer a la espera durante otro periodo de tiempo.

En este ensayo la frecuencia de muestreo ha sido de 2 KHz y tanto el tiempo de muestreo como el tiempo en espera de 2 s.

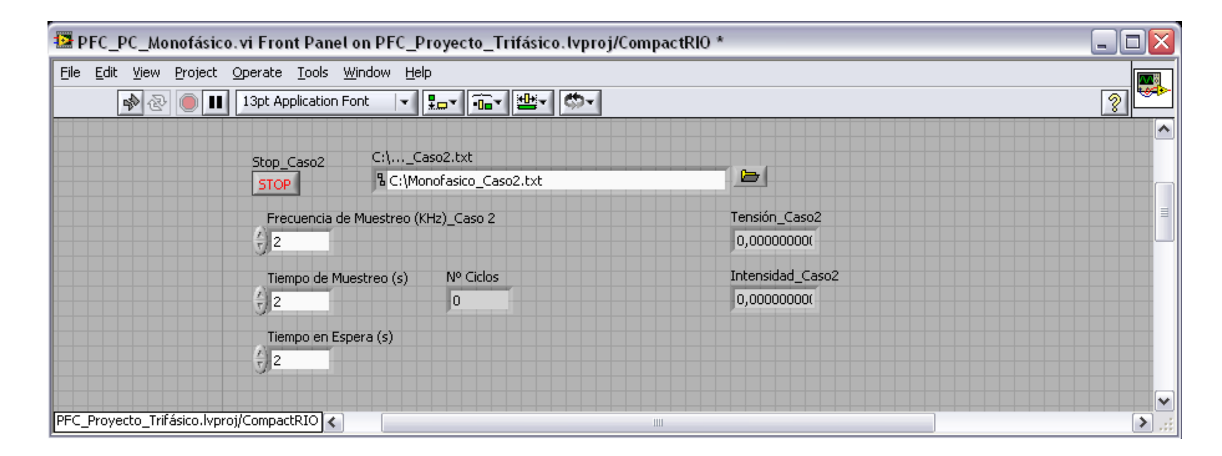


Figura 66: Panel Frontal_Ensayo caso 2_Ensayo de toma automática de datos.

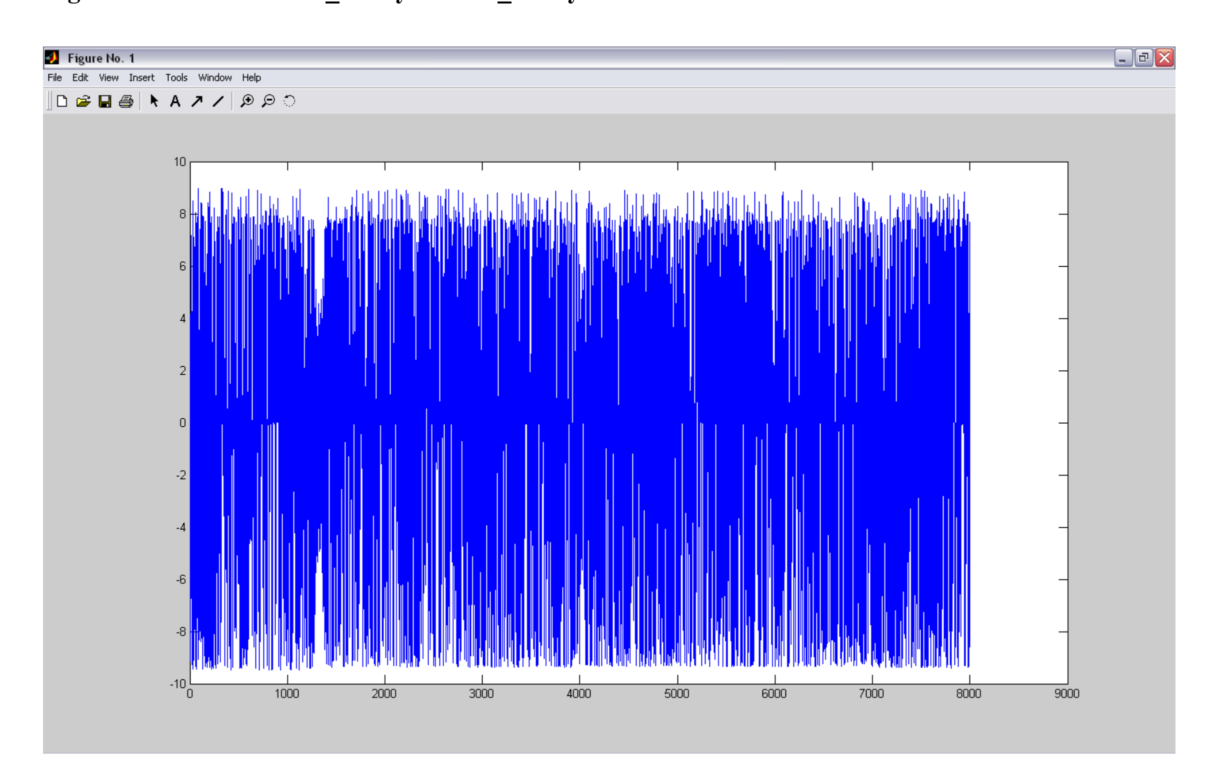


Figura 67: Ensayo caso 2_Ensayo de toma automática de datos.

Como se puede observar en este ensayo hemos obtenido unas 8.000 muestras.

Ensayos de huecos de tensión

Grabar un buffer de datos cuando se cumpla una condición especificada por el usuario.

Ensayo Monofásico

Parámetros en este ensayo son:

Frecuencia de muestreo = 10 KHz
Tiempo antes del evento = 0,2 s
Tiempo después del evento = 0,2 s
Frecuencia de la red = 50Hz
Umbral % de RMS <… = 90%
RMS_nominal = 231,8 V
Tensión Primario = 231,8 V
Tensión Secundario = 4,68 V

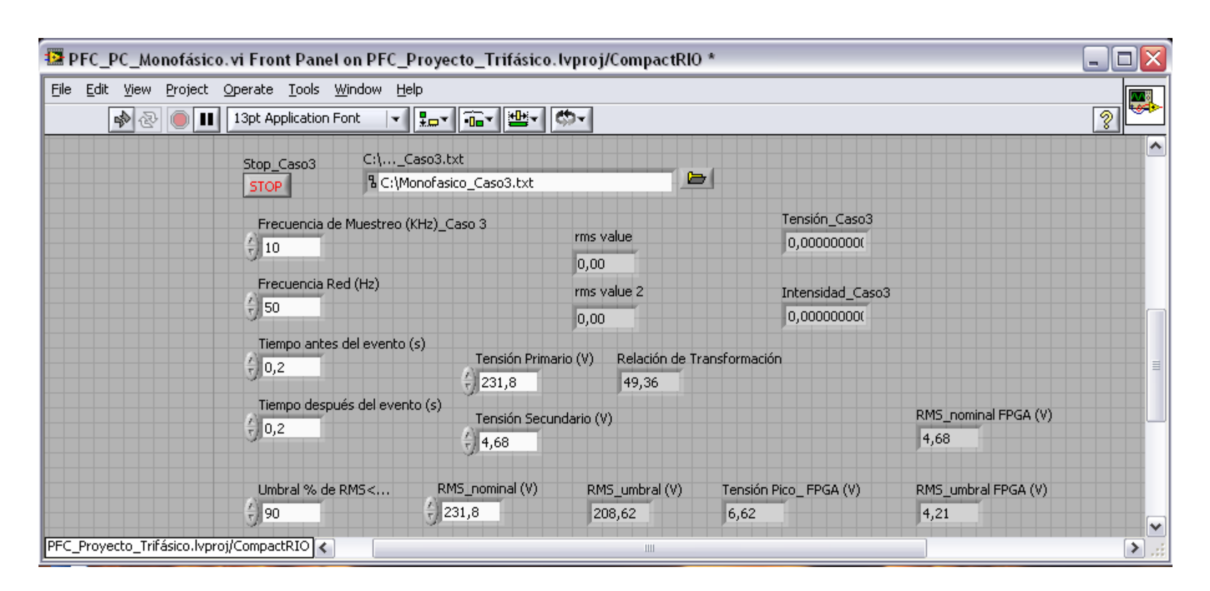


Figura 68: Ejemplo 1_Panel Frontal_Ensayo caso 3_Hueco de tensión_Monofásico.

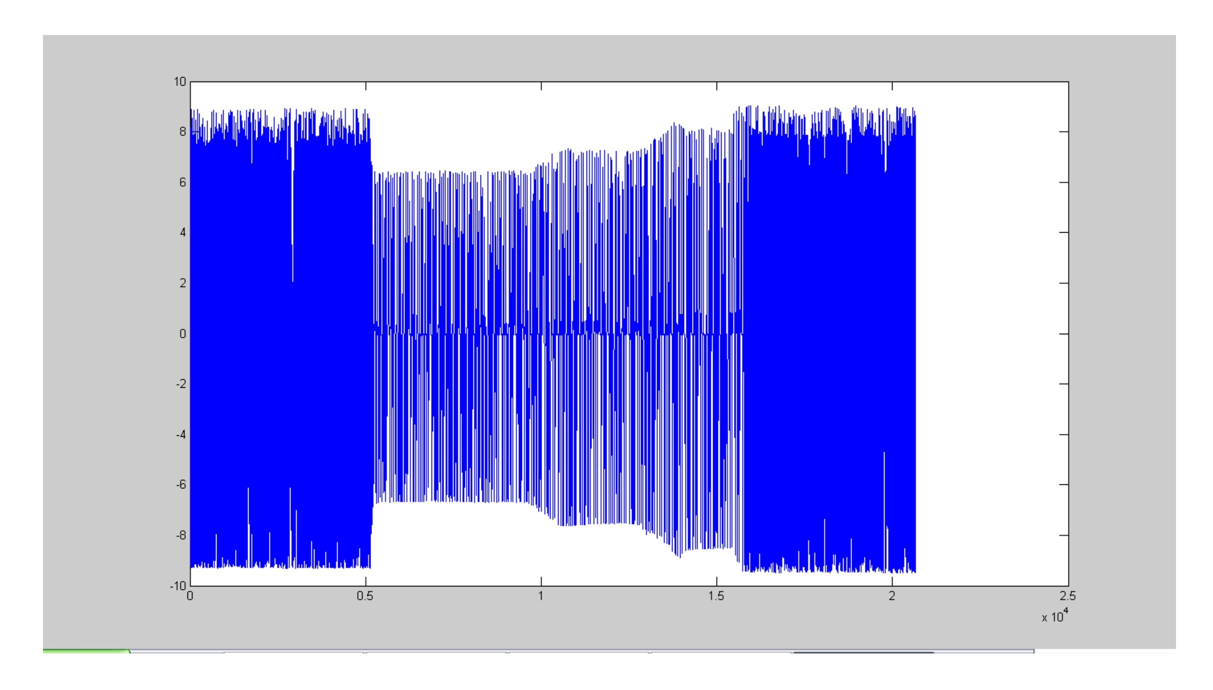


Figura 69: Ejemplo 1_Ensayo caso 3_Hueco de tensión_Monofásico.

Se puede observar en la Figura 69 que se ha detectado un hueco de tensión correspondiente a un gran número de ciclos de onda, es decir, lo que hemos denominado ciclos evento.

Ensayo Monofásico

Parámetros en este ensayo son:

Frecuencia de muestreo = 10 KHz
Tiempo antes del evento = 0,5 s
Tiempo después del evento = 0,5 s
Frecuencia de la red = 50Hz
Umbral % de RMS <… = 90%
RMS_nominal = 231,3 V
Tensión Primario = 231,3 V
Tensión Secundario = 4,65 V

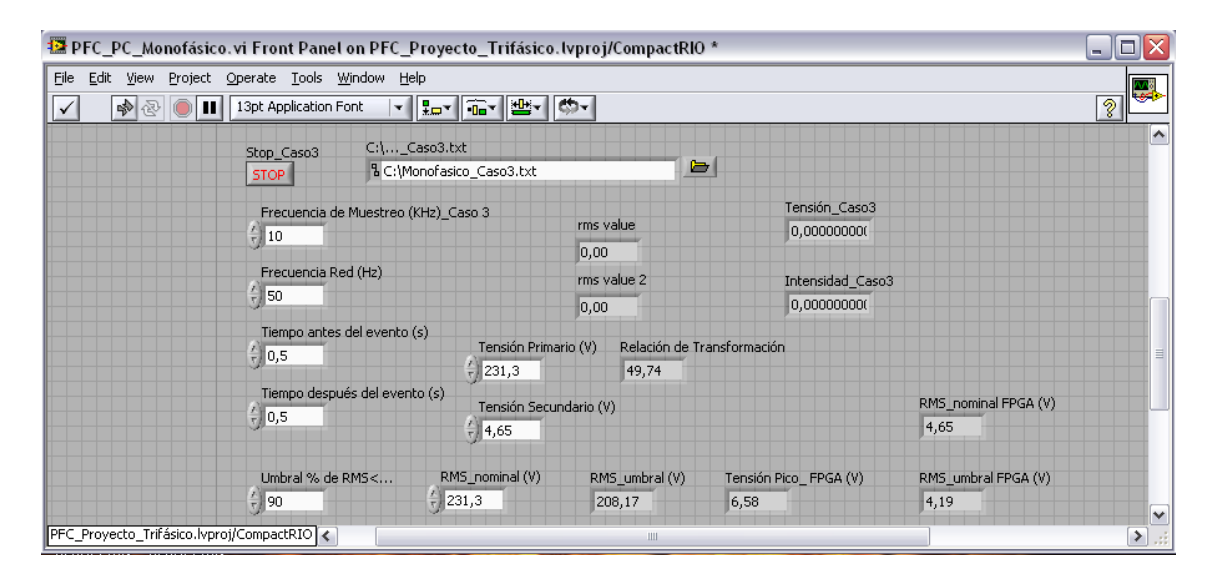


Figura 70: Ejemplo2_Panel Frontal_Ensayo caso 3_Hueco de tensión_Monofásico.

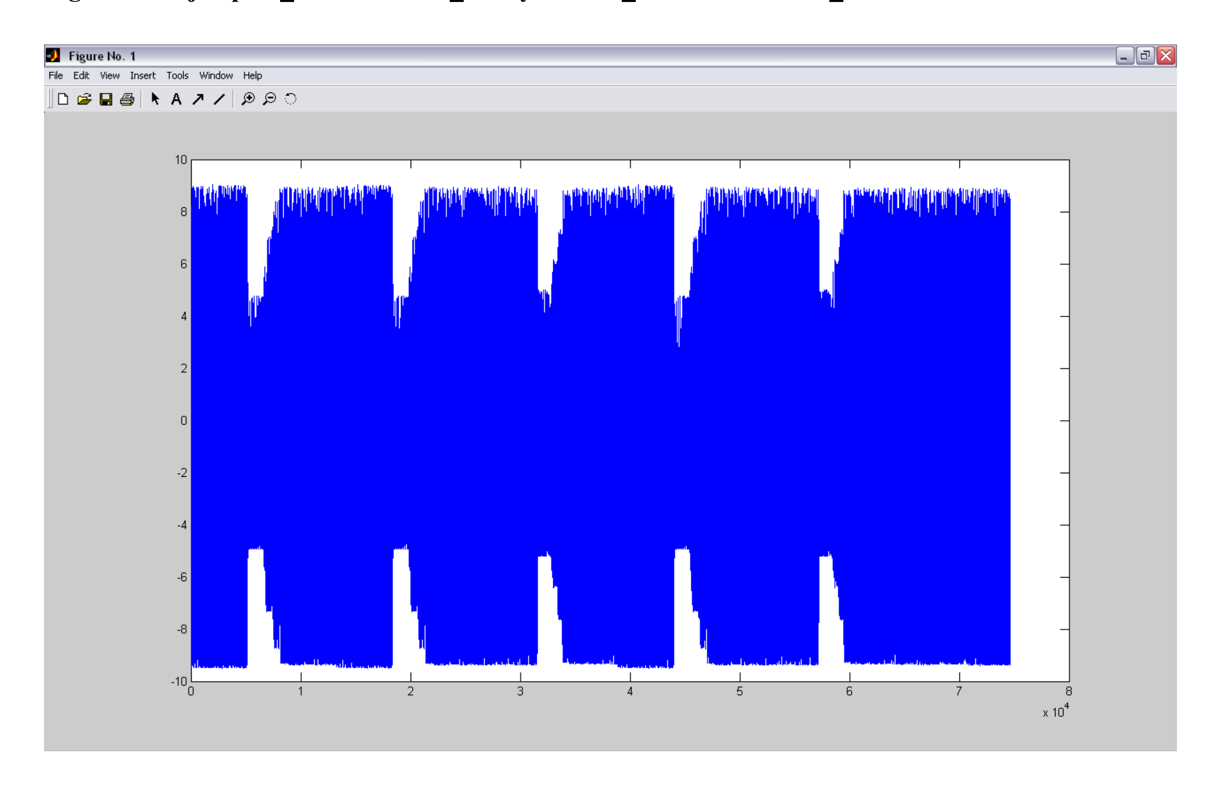


Figura 71: Ejemplo2_Ensayo caso 3_Hueco de tensión_Monofásico.

Como se puede ver en este caso (Figura 71) hemos detectado cinco huecos de tensión.

Ensayo Trifásico

Parámetros en este ensayo son:

Frecuencia de muestreo = 5 KHz
Tiempo antes del evento = 1 s
Tiempo después del evento = 1 s
Frecuencia de la red = 50Hz
Umbral % de RMS <… = 75%
RMS_nominal = 394,1 V
Tensión Primario = 394,1 V
Tensión Secundario = 4,30 V

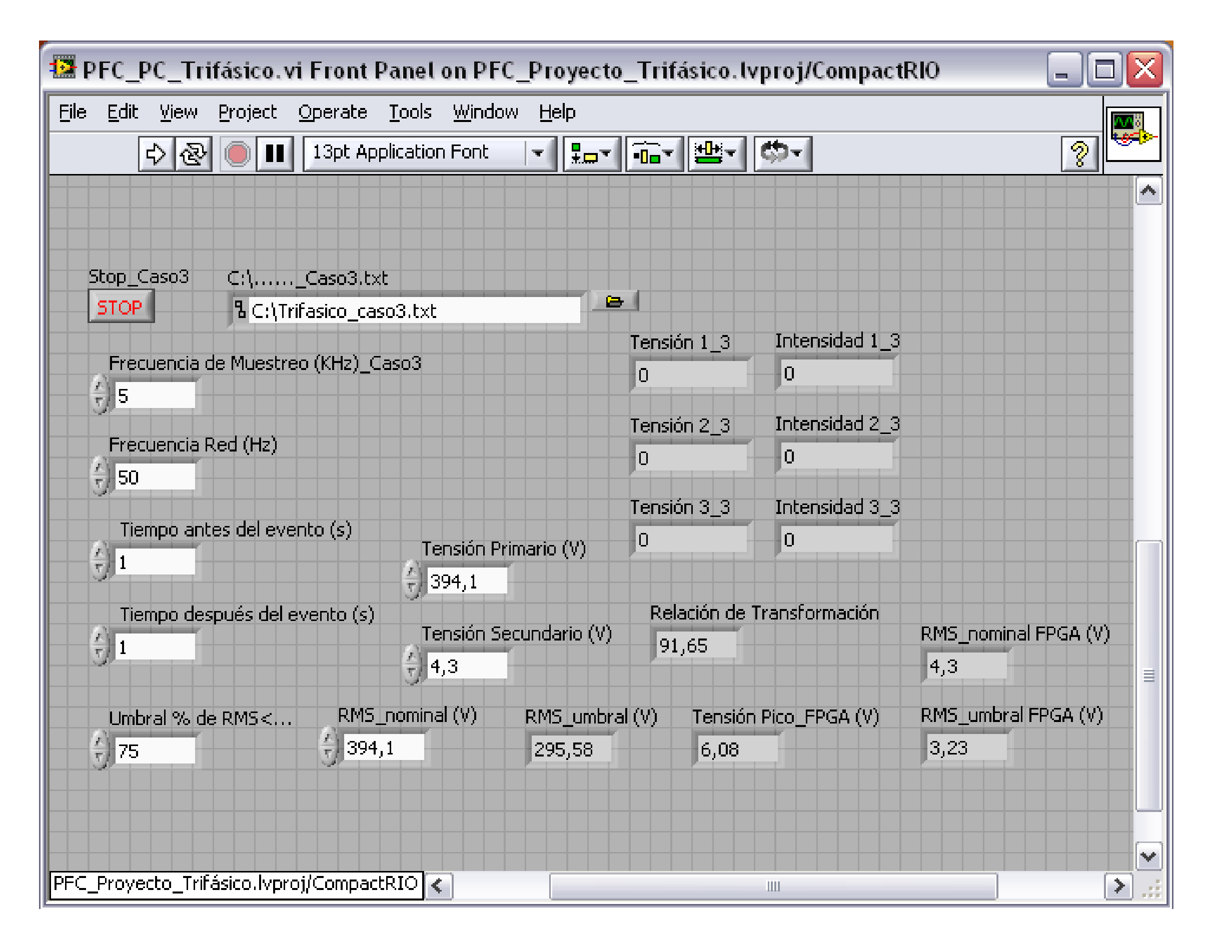


Figura 72: Panel Frontal_Ensayo caso 3_Hueco de tensión_Trifásico.

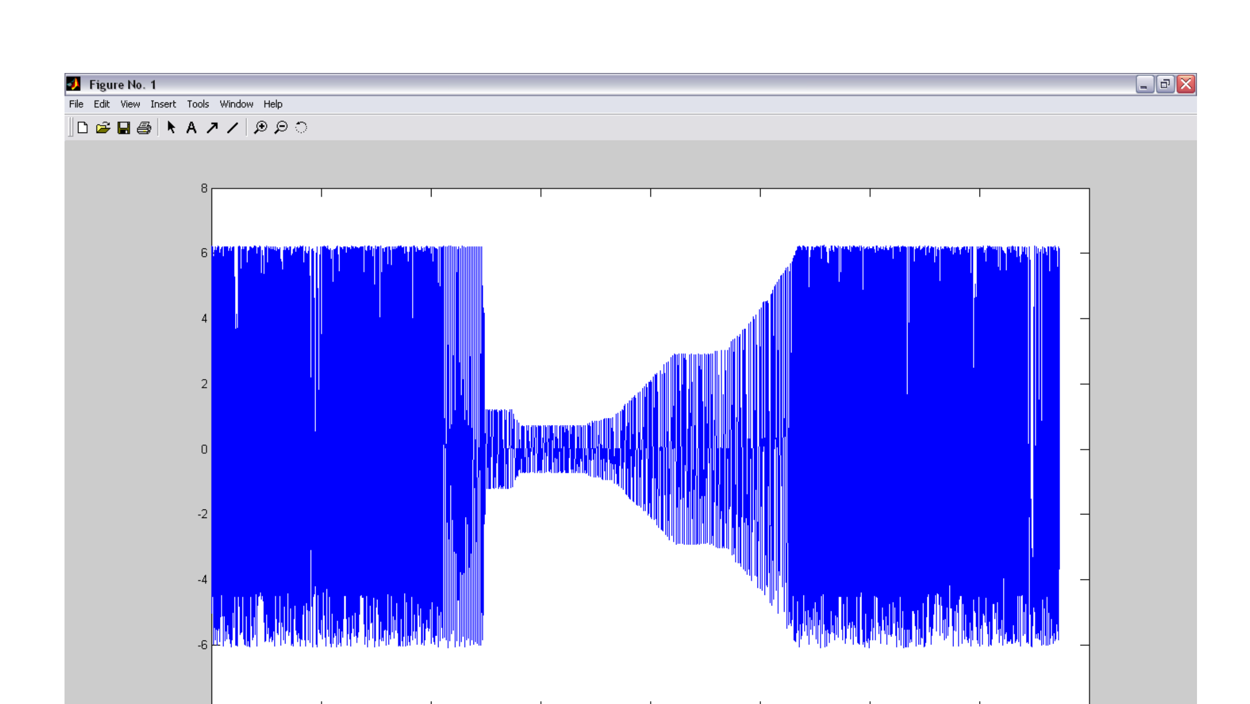


Figura 73: Ensayo caso 3_Hueco de tensión_Trifásico_Canal Tensión 1.

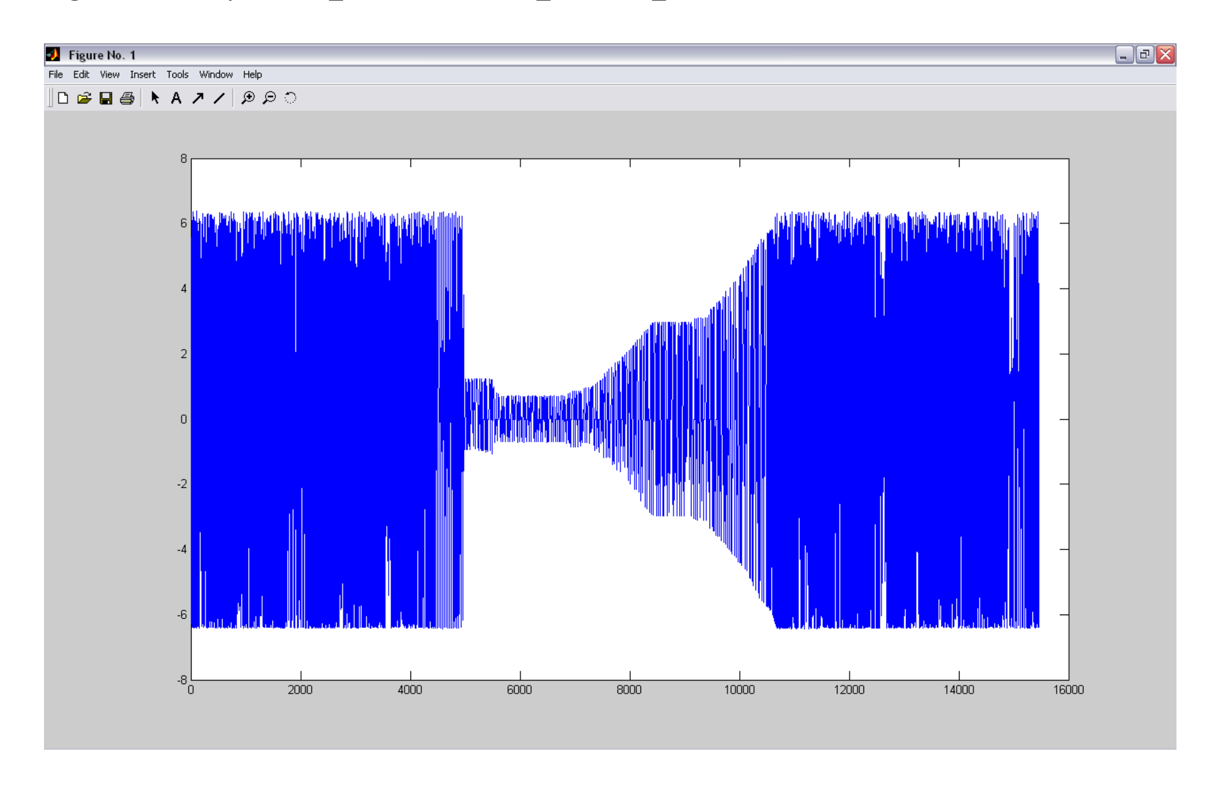


Figura 74: Ensayo caso 3_Hueco de tensión_Trifásico_Canal Tensión 2.

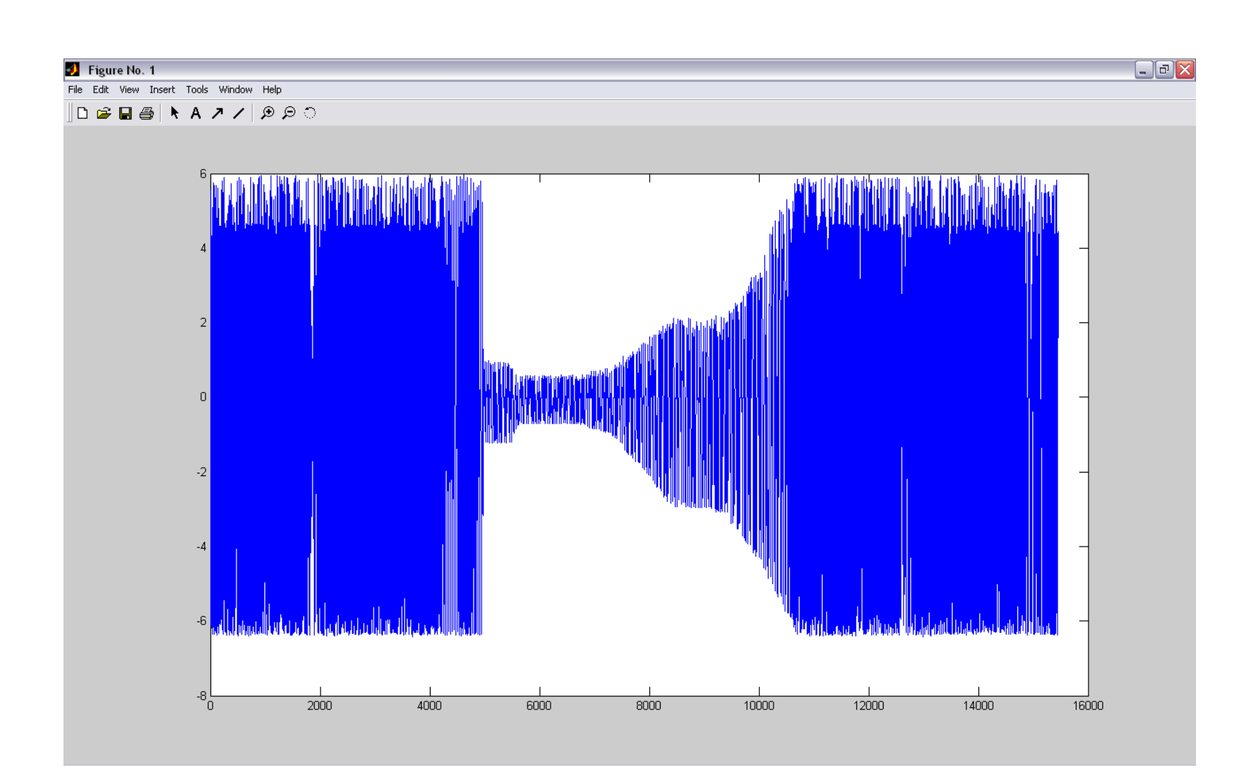


Figura 75: Ensayo caso 3_Hueco de tensión_Trifásico_Canal Tensión 3.

Donde se puede observar la aparición de un hueco de tensión. En la siguiente figura (Figura 76) hemos hecho un zoom en el hueco de tensión para observar la onda mejor.

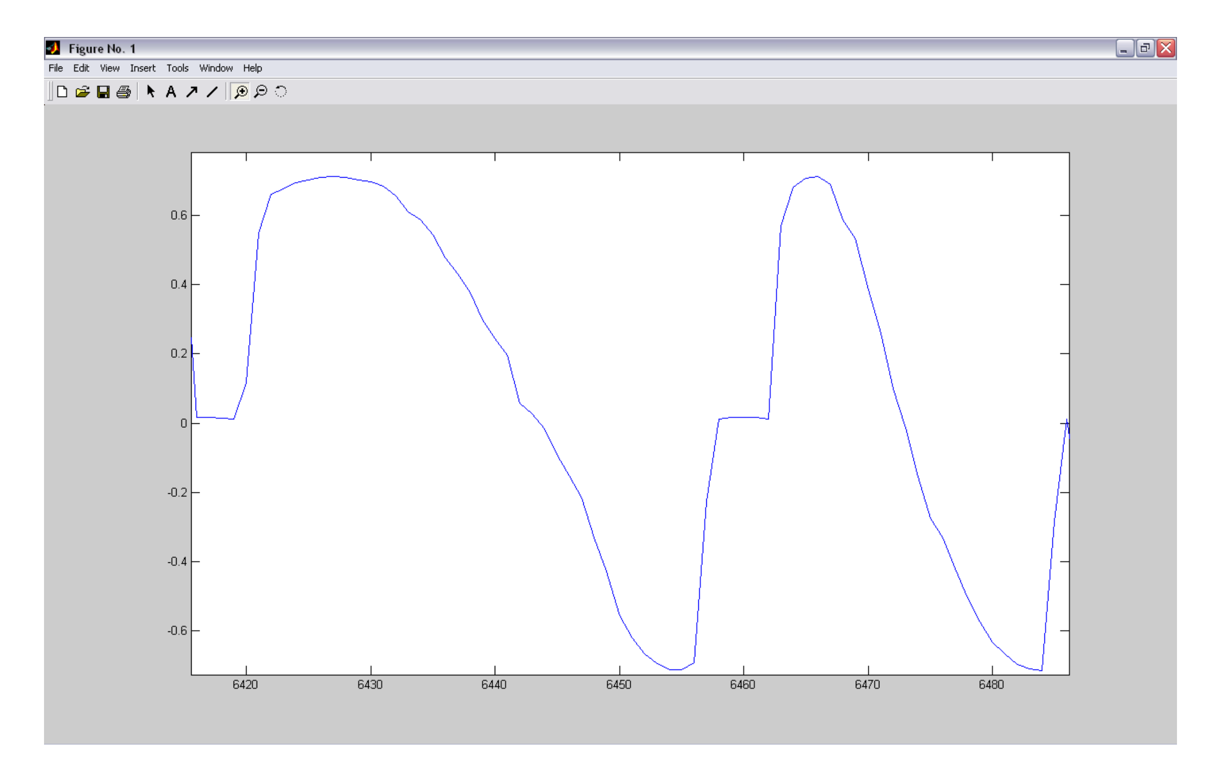


Figura 76: Zoom en el hueco de tensión.

Conclusiones y trabajos futuros

Como hemos podido comprobar el CompactRIO ha cumplido con todos nuestros objetivos. Es capaz de detectar los huecos de tensión, como también de adquirir datos con una suficiente frecuencia y con una capacidad más que aceptable para su grabación.

Debido a la gran flexibilidad que presenta el CompactRIO se podría utilizar para el estudio de cualquier otro tipo de eventualidad como la detección de sobreintensidades. Simplemente consistiría en modificar parte del código de programación utilizado en esta aplicación.

Bibliografía

- LabVIEW. Entorno gráfico de programación. LabVIEW 8.20 y versiones anteriores. Ed. marcombo ediciones técnicas 2007. Autores: José Rafael Lajara Vizcaíno y José Pelegrí Sebastiá.
- LabVIEW. Data Acquisition and Signal Conditioning Course Manual. NI Customer Education. Curse Software Version 8.0 October 2005 Edition. Part Number 320733L-01. National Instruments.
- Introducción a LabVIEW. Curso 4 Horas 2005. National Instruments.
- Introducción a LabVIEW. Curso de Seis-Horas. September 2003 Edition. Part Number 323669B-01. National Instruments.